

PLANNING COLLISION-FREE PATHS WITH
APPLICATIONS TO ROBOT MANIPULATORS

By

IHN NAMGUNG

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

1989

Copyright 1989

by

Ihn Nangung

In memory of Namgung, Seob
(1911-1987)

ACKNOWLEDGEMENTS

The author wishes to thank his advisor Dr. Joseph Duffy for supporting him at the early stage of this work, and guiding and shaping him as an independent researcher. He extends his sincere gratitude toward the remaining committee members: Dr. Carl Crane, Dr. Ali Seireg, Dr. Thomas Bullock and Dr. Ralph Selfridge for their advice and many comments.

The author also would like to thank his fellow graduate students in CIMAR (Center for Intelligent Machines And Robotics), especially Paul Goodall and Chris Kardish for numerous discussions in writting the work.

Above all, the author would like to thank his parents, Kyun Namgung and Imhwa Chung, and wife, Yearnok for their love, support and encouragement, without which this work could not have been completed.

TABLE OF CONTENTS

	<u>page</u>
ACKNOWLEDGMENTS.....	iv
ABSTRACT.....	vii
CHAPTERS	
1 INTRODUCTION.....	1
2 SURVEY OF LITERATURE.....	8
2.1 Path Planning and Collision Avoidance.....	8
2.2 Mobile Robots.....	21
2.3 Concluding Remarks.....	24
3 CONCEPTUAL DEVELOPMENT OF COLLISION-FREE PATHS.....	26
3.1 Parametric Curves.....	27
3.2 Collision-Free Paths Connecting Two Designated Points in an Obstacle Strewn Two Dimensional Environment....	42
3.3 Collision-Free Paths Connecting Two Designated Points in an Obstacle Strewn Three Dimensional Environment..	51
3.4 Concluding Remarks.....	55
4 ANALYSIS OF COLLISION-FREE PATHS.....	56
4.1 Straight Paths.....	56
4.2 Geometric Mapping of Two Dimensional Obstacles.....	61
4.3 Composite Paths.....	76
4.4 Three Dimensional Collision-Free Paths.....	87
4.5 Implementation of the Algorithms.....	91
4.6 Concluding Remarks.....	93
5 COLLISION AVOIDANCE OF INDUSTRIAL ROBOTS IN CONJUNCTION WITH PLANNING COLLISION-FREE PATHS.....	96
5.1 Expansion of Obstacles.....	99
5.2 Collision Recognitions.....	103
5.3 Change of Configurations.....	108
5.4 Planar 3R Robots.....	110
5.5 General Electric P60 Robots.....	119
5.6 Concluding Remarks.....	125

	<u>page</u>
6 CONCLUSIONS.....	128
6.1 Planning Collision-Free Paths.....	128
6.2 Collision Avoidance of Robot Manipulators.....	129
6.3 Further Considerations.....	129
APPENDICES	
A SIMULATION RESULTS OF PLANAR COLLISION-FREE PATHS.....	132
B DISPLACEMENT ANALYSIS OF THE GE P60 ROBOT.....	144
B.1 Reverse Displacement Analysis.....	146
B.2 Forward Displacement Analysis.....	150
REFERENCES.....	152
BIOGRAPHICAL SKETCH.....	163

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

PLANNING COLLISION-FREE PATHS WITH
APPLICATIONS TO ROBOT MANIPULATORS

By

IHN NAMGUNG

DECEMBER 1989

Chairman: Dr. Joseph Duffy
Major Department: Mechanical Engineering

A new approach for planning a feasible collision-free path based on a Bézier curve of order two, a parabola, has been developed for two and three dimensional environments. It is assumed that obstacles are stationary and the shape and location of obstacles are known. The algorithm involves geometric analysis and produces a path connecting a user specified start and target point which does not interfere with any obstacles. The algorithm consists of three parts: a geometric mapping of obstacles, selection of intermediate points, and smoothing of the final path.

A geometric mapping transforms obstacles in Euclidean space into obstacle images in the parameter space (θ, t) . The main property of the geometric mapping is that a point in the parameter space which is not mapped by obstacle images, corresponds to a collision-free parabola in Euclidean space. If the entire parameter space is occupied by obstacle

images, intermediate connection points are searched. Selections of intermediate connection points are made near the vertices of the obstacle that blocks the straight line connecting the start and target point. Geometric mapping is then carried out for each subinterval, and smoothing the slope discontinuity at each connection point is performed to ensure the overall smoothness of the curve.

Three supporting algorithms--expansion of obstacles, collision recognition, and change of configuration without changing the end-effector position--are developed for collision avoidance of a planar 3R robot and a General Electric P60 robot. Expansion of obstacles is introduced to accommodate the size of the end-effector. Collision recognition of robot links is carried out by checking interferences among robot links and obstacles. Change of configuration by rotating orientation of the end-effector, without altering the position, is introduced to alleviate the problems associated with an attempt to move out of the reachable workspace, exceed the valid range of joint-actuators or collide with obstacles.

Animated simulation of collision-free path generation and collision avoidance of a planar 3R robot and a General Electric P60 robot is implemented on a Silicon Graphics 4D/70 graphics workstation.

CHAPTER 1 INTRODUCTION

The key characteristic of a robot system is that it can be programmed to perform many different tasks. Robot programming can be divided into three levels: task-level control, environment coordination, and manipulator control. These levels are extensions of those described by Lozano-Pérez (1982). Task-level control is the highest level motion specification. For example, a user specifies that a pin should be placed in a hole rather than specifying the sequence of manipulator motions needed to perform the insertion. World modelling, path planning, grasp planning, collision avoidance and sensory feedback comprise the environment coordination level. Manipulator control is directly responsible for controlling the robot; manipulator kinematics, dynamics and joint actuator control belong to this level. Successful task-level control requires a robust environment coordination scheme, thus the widespread use of industrial robots has been hampered by the insufficient implementation of this intermediate level of control. Path planning and collision avoidance are studied here as a step toward for overcoming these inadequacies.

In this dissertation, algorithms are developed for real-time planning of a feasible collision-free path in two and three dimensional environments. It is assumed that all obstacles are stationary and fully

known. Collision-free path here refers to a series of points joining two designated points without any interfering obstacles. The algorithms for planning collision-free paths are based on planar Bézier curves of order two, parabola, and geometric analysis. Smoothness of the path, specifically continuity in the first derivative, is considered in the process of path planning. The motivation for using a Bézier curve of order two is the existence of a closed form solution for the geometric mapping. Closed form solutions are not available from higher order curves. The geometric mapping transforms obstacles in Euclidean space to parameter space; unoccupied points in the parameter space (θ, t) correspond to collision-free paths in Euclidean space.

In general, there are two types of problems in collision avoidance. In the first a robot is guided along a user specified path while avoiding collisions with obstacles in which the path is generated separately. In the other the start and target point are given and a robot maneuvers to the target point while avoiding collisions with obstacles in which the path is found in the joint space of the robot. The first case can be found in Young and Duffy (1987) and it is computationally effective. The second case can be found in Brooks (1983b) and it appears that the method is computationally prohibitive for robots having more than three joints. The first case is considered here and a path is provided by the above path planning algorithms. Algorithms are developed for expanding obstacles to accommodate the size of the end-effector and links as part of path planning and collision detection respectively. Other algorithms are developed for detecting collisions for a planar 3R robot and a General Electric P60 robot. Algorithms are also developed for avoiding the limit

cases such as collision detection, joint limits or reachable workspace limits of a planar 3R robot and a General Electric P60 robot by changing the configuration without altering the end-effector position. Change of configuration by changing both orientation and position of the end-effector is not considered here.

A comprehensive survey of path planning, collision avoidance for robot manipulators and mobile robots is presented in Chapter 2. The review of mobile robots is included because many of the path planning and collision avoidance schemes have been investigated for implementation with mobile robots. These algorithms for planning collision-free paths include the visibility graph search, free space search, potential field method, bug's algorithm and others. In the visibility graph search, connections are made among vertices of obstacles and start and target points. The connections that do not intersect any obstacle are organized in a tree structure. The shortest path is found by search methods established from AI researches. In the free space search, the workspace is divided into several cells and obstacle occupancy of the cells is determined as empty, mixed and full. Mixed cells are further divided into smaller cells and occupancy is determined again. Empty cells are organized in a tree structure and a connection of adjacent empty cells from start to target point results in a path. In the potential field method, obstacles are represented by repulsive poles and the target point is represented by an attractive pole. A collision-free path from start to target point is obtained by optimization methods. A method resembling a bug's movement connects start and target points by straight line segments and boundaries of obstacles. A bug follows the straight path toward the target point.

If an obstacle blocks the path then it follows the boundary of the obstacle until it returns to the straight path. Other schemes include a method utilizing Snell's law. Obstacle avoidance of robot manipulators is divided into two classes of problems. One assumes that a path is given, and that the robot maneuvers to avoid collision while moving along the path. The other assumes that the path is generated in joint space; as a result no obstacle avoidance is necessary. Some of the notable simulations on collision avoidance include the work by Udupa (1977), Brooks (1983b), Maciejewski and Klein (1985) and Young and Duffy (1987). Collision avoidance of a mobile robot may be attained simply by expanding obstacles and planning collision-free paths.

Conceptual development of collision-free paths is presented in Chapter 3. Fundamentals of parametric curves essential to this development are included in section 3.1. The geometric mapping, based on Bézier curves, which transforms obstacles in Euclidean space to the parameter space is introduced in sections 3.2 and 3.3. An example of the geometric mapping for a two dimensional case is shown in Figures 1-1 and 1-2. The control point associated with a Bézier curve is defined by the two independent parameters (θ, t) . A selection of (θ, t) in the clear area of parameter space results in a collision-free path.

Detailed analysis for the algorithms of planning collision-free paths is described in Chapter 4. The geometric mapping consists of the determination of parameter range, calculation of intersection between parabolas and obstacles and selection of a point from the clear area of parameter space. When the entire parameter space is occupied by images of obstacles, a collision-free parabola is not available for the given

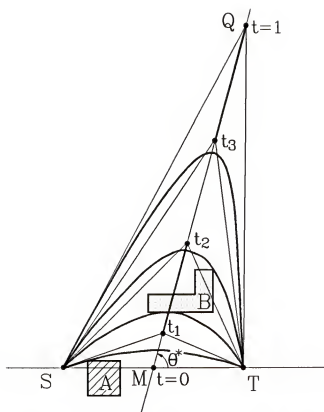


Fig. 1-1 Obstacles in Euclidean space

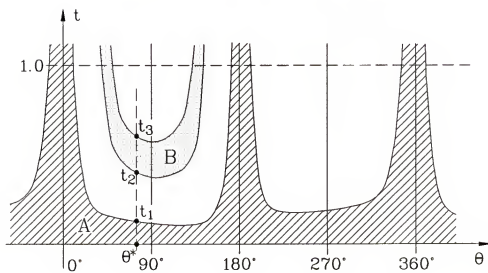


Fig. 1-2 Obstacle images in parameter space

interval. Intermediate connection points are searched near the vertices of the obstacles which intersect the straight line connecting the start and target points. After finding intermediate connection points, the geometric mapping for each subinterval determined above is carried out and the slope discontinuity at the connection points is then smoothed with a higher order curve. In the case of three dimensional geometric mapping, the parameter space is defined by three parameters (ϕ , θ , t), and the parameter ϕ is assigned first to define the projection plane. The images on the projection plane are obtained by intersections between obstacles and the projection plane. Then the geometric mapping developed for two dimensional cases is applied to the images on the projection plane. If a projection plane does not produce a collision-free path, the parameter ϕ is changed and the above processes are carried out again until a successful path is found. Some of the results of simulation for the two dimensional collision-free paths are given in Appendix A.

In Chapter 5, collision detection and change of configuration of a robot manipulator are developed as part of collision avoidance of a planar 3R robot and a General Electric P60 robot. As mentioned previously, collision avoidance considered in Chapter 5 is that a robot is guided along a given path while avoiding collisions. It is assumed that the orientation of the end-effector is free to change and the position cannot be altered during the change of configuration. Collisions among robot links and obstacles are identified by checking intersection of links with obstacles. Robot links are represented by line-segments to reduce necessary computation for interference, and obstacles are expanded to accommodate the size of link. As the robot moves along the path with

its initial orientation, the limit cases, described previously, are monitored. Should such an event occur, configuration of the robot is changed in real-time while staying at the same position by simply rotating the orientation of the end-effector. For a General Electric P60 robot, several rotation axes are pre-selected and the actual rotation occurs about the axis that provides the most significant change in orientation. Algorithms for collision detection can be augmented in existing robot trajectory generators, thereby reducing the robot programming time.

Animated simulations of collision-free path planning, and collision detection and change of configuration for a planar 3R robot and a General Electric P60 robot are presented on a Silicon Graphics IRIS 4D/70 graphics workstation using the C programming language. Some of the results are given in Chapter 5.

CHAPTER 2

SURVEY OF LITERATURE

The survey is focused on path planning, obstacle avoidance and mobile robots. Mobile robot strategies resemble obstacle avoidance for a point. It is an attempt to describe the current state of research in these fields.

2.1 Path Planning and Collision Avoidance

The algorithms for robot path planning and collision avoidance can be grouped into the following categories:

1. Visibility graph search,
2. Free space search,
3. Potential field method,
4. Bug's algorithm,
5. Others.

The work by Udupa (1977) is regarded as one of the first for collision detection and avoidance of a manipulator. He used the Scheinman arm, which has a prismatic joint for the third link. Collision-free path planning was carried out in a two level hierarchy: path planning for the boom, which is the third link, was done first, then obstacle avoidance for the wrist joint was done. Path planning for the boom was regarded as a planar case in which free space was characterized by circular sectors.

Path planning was done successively from start position to goal position through free circular sectors.

Visibility graph search method is one of the early approaches for planning collision-free paths. Lozano-Pérez and Wesley (1979) introduced the idea of visibility graph search. Visibility graph (VGRAPH) is a set of all connected links among vertices of obstacle images in configuration space, including start and goal position, that do not intersect with obstacles. The shortest collision-free path from start to goal is the shortest in the VGRAPH from the node corresponding to start to that corresponding to goal when the Euclidean metric is used on the links in the VGRAPH, see Fig. 1-1. An A^* graph search is used to find the shortest path from the VGRAPH. A^* is a classical minimum cost graph search algorithm, whose optimality properties are well known as shown by Nilsson (1980). The algorithm requires the moving object to be a point. If the moving object is not a point, a new set of obstacles must be computed. The operation of computing a new obstacle from an original obstacle, called a grown obstacle, depends on the movement of the object. If translation is allowed and rotation is not allowed, the grown obstacle operation is carried out by expanding edges of the original obstacles, Fig. 2-2.

Configuration space of an object is defined by the position and rotation of the body in Euclidean space. As the freedom of the object increases, the VGRAPH cannot be applied in a simple manner. Even for a planar case the configuration space is three dimensional: translation (x, y) and rotation (θ). In order to reduce computing time the rotational parameter (θ) is sliced into many small angles and the VGRAPH algorithm

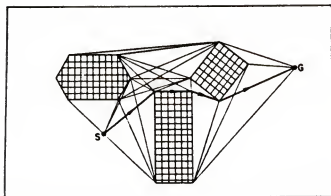


Fig. 2-1 Visibility graph, (from Lozano-Pérez and Wesley, 1979)

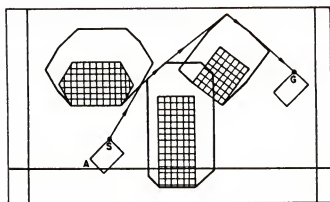


Fig. 2-2 Grown obstacles, (from Lozano-Pérez and Wesley, 1979)

is applied for each slice. Since the VGRAPH algorithm does not take into account the motion of the robot body, it is applicable only for the cartesian robots in which the translational degrees of the robot are separated from the rotational. Lozano-Pérez (1981, 1983) suggested a swap volume method to represent the rotational links and the gripper of the cartesian robot as a bounding box.

A rigorous mathematical treatment on the "Piano Movers" problem was given in a series of papers by Schwartz and Sharir (1983a, 1983b, 1983c, 1984), and Sharir and Ariel-Sheffi (1984). The problem is:

Given two configurations (i.e., positions and orientations of all subparts) of the bodies B in which none of these bodies touches any walls, and in which none of the bodies B collide, find a continuous wall- and collision-avoiding motion of all the B between these two configurations, or establish that no such motion exists. Schwartz and Sharir (1983b, p.299).

Moving a two dimensional polygonal body amidst polygonal obstacles was given in the first paper of the series. The algorithm is polynomial in the number of walls ($O(n^5)$ ¹ if n is the number of walls). In the second paper the case of arbitrary number of moving bodies, some of which may be jointed, was treated. It was shown that the problem can be solved in time that is polynomial in the number of the walls and the bodies, but exponential in the number of degrees of freedom of the system of bodies. However, even for a fixed number of degrees of freedom, the algorithm presented is of complexity $O(n^e)$, where the exponent e can be high. As

¹ $O(n^5)$ indicates that the number of geometric operations, such as intersection of a point with a line-segment, required to carry out the algorithm is at most as large as n^5 .

a result, this general algorithm is entirely impractical except possibly for the simplest cases.

In the third paper, each obstacle was enclosed within a circle, and then a path is planned for the motion of these circles.¹ A more precise algorithm, taking into account the exact geometry of the moving bodies, will be brought to bear only if no collision-free motion of the enclosing circle exists. The problem solved in the fourth paper was that of planning motions for a two dimensional robot system consisting of several arms all jointed at one common endpoint and free to rotate past each other. The algorithm was polynomial in the number of wall edges $O(n^{k+4})$, where k is the number of arms. Planning the motion of a rod moving amidst polyhedral objects is treated in the fifth paper.

The main strategy used in the analysis of the 'piano movers' problem was an investigation of free configuration space, FP, where a body (or group of related bodies) B is constrained to move freely in a two or three dimensional space bounded by certain obstacles. The dimension n of configuration space is decomposed into two subproblems of size 1 and $n-1$, or equivalently projected onto a subspace of lower dimension $n-1$. This decomposition is repeated until the dimension n is reduced to one or two. A connectivity graph for the vertices of each obstacle is formed between the projection. The advantage of this technique is that it reduces an n -dimensional problem into decomposition problems involving manifolds of lower dimensions. However, each step of projection complicates significantly the geometric structure of the manifold to be analyzed, and

¹ Such simplification can also be found in Moravec(1980).

so recursive application of the technique becomes difficult after two or three levels of recursion.

The work by Bajaj and Kim (1987, 1988) is concerned with effective construction of configuration space. In their first paper a planar case with translation only was treated, and in the second paper configuration space for a sphere was treated where the problem was reduced to planning motion for a point among grown obstacles.

Note that the configuration space is actually the same as joint space. For a 6R industrial manipulator the joint space is 6-dimensional, and the configuration space of each link is 6-dimensional. However, the constraint imposed on each joint is 5-dimensional; hence, the configuration space for the manipulator is $6 \times 6 - 5 \times 6 = 6$. One of the characteristics of the free configuration space which contains no obstacles is that it guarantees collision avoidance.

One of the most successful implementations of the collision avoidance algorithm is found in Brooks and Lozano-Pérez (1985) where a simulation of moving an object through a free space was given. Gouzénes (1984) presented an implementation of planning collision-free path algorithms based on Lozano-Pérez's method. An implementation by Lozano-Pérez (1986, 1987) is motion planning for simple robot manipulators, where the first three joints were treated as active. Brooks and Lozano-Pérez (1985) reported that most of the computing time was spent during the A^* search.

A simple method of finding collision-free paths was introduced by Brooks (1983a). Free space is represented as overlapping generalized cones, Fig. 2-3. Generalized cones are constructed by examining all pairs

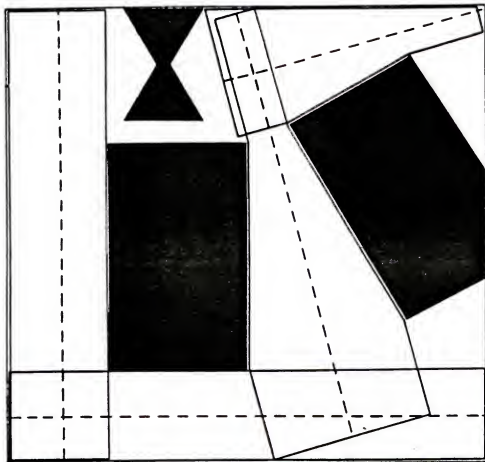


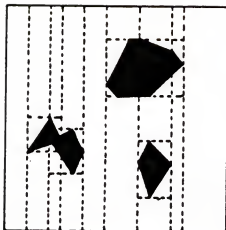
Fig. 2-3 Generalized cones generated by two obstacles and workspace boundary, (from Brooks, 1983a)

of edges of obstacle polygons. Two edges are accepted as defining a candidate generalized cone if they meet the two requirements: 1) at least one vertex of each edge should be on the outside of the other, 2) the dot product of the outward pointing normals should be negative. The generalized cone so defined may not lie entirely in free space; there may be other obstacles which intersect it. Each obstacle is compared to the

generalized cone. If the intersection occurs, a modification of the generalized cone is done. The center line of a generalized cone is a spine. The cones are examined to determine if and where their spines intersect. Each spine intersection point is examined for orientations of the object A in which it is guaranteed to be completely contained in the generalized cone. A connectivity graph is built based on spine intersection, and A* search is used to find a collision-free path.

In Brooks and Lozano-Pérez (1985), the configuration space is divided into rectangles, Fig. 2-4. Each rectangloid is labeled as 1) empty if the rectangloid does not intersect a configuration obstacle, 2) full if the rectangloid is contained within a configuration obstacle, or 3) mixed if the rectangloid intersects a configuration obstacle boundary. A free path is searched by a connected set of empty rectangloid cells from starting position to goal position. If an empty cell path cannot be found in the initial subdivision of configuration space, then a path that includes mixed cells is searched. Mixed cells on the path are further subdivided and the resulting cells are labeled as empty, full or mixed. Another search for an empty cell path for the subportion is initiated. Gouzenes (1984) approximated the decomposition of the configuration space by regular grids. A cell is empty if no obstacles are inside the cell; otherwise a cell is full. All the empty cells are represented in a tree structure. A path is found by searching through the tree. Note that it is not necessary to carry out the decomposition in the configuration space.

A grid representation of workspace was given by Jun and Shin (1988) and Kambhampati and Davis (1986). In Jun and Shin (1988), the workspace



(a) Configuration space without rotation of obstacles cut into rectangular cells



(b) Connectivity graph for cells in (a)

Fig. 2-4 Configuration space represented by rectangular cells and associated visibility graph, (from Brooks and Lozano-Pérez, 1985)

was represented by regular cubes, and in Kambhampati and Davis (1986), it was represented by a quadtree. A quadtree is a recursive decomposition of a two dimensional workspace into uniformly sized $2^i \times 2^i$ blocks, Fig. 2-5, where i is the depth of a quadtree. A node of a quadtree represents

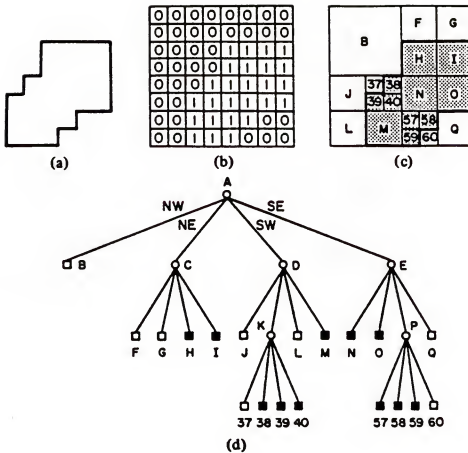


Fig. 2-5 Quadtree representation, (from Kambhampati and Davis, 1986)

a $2^i \times 2^j$ square region. A free node of a quadtree is a node of the quadtree representing a region of free space. An obstacle node is a node representing a region of obstacle. A gray node is a node representing a region having a mixture of free space and obstacles. A leaf node of a quadtree is a tip node of the tree. A collision-free path is a connection of free nodes.

Grid based path planning can also be found in Mitchell and Keirsey (1984). The possible connections between nodes are diagonal, horizontal and vertical. The search for a collision-free path is, therefore, much simplified. Singh and Wagh (1987) and Wu et al. (1987) also used a grid map to represent the workspace. The grid map depends on the size of obstacles, and obstacle edges are assumed to be aligned with coordinate axes. The work by Elfes and Matthies (1987) is sonar and stereo sensor integration. The workspace is represented by an occupancy grid map using sonar and stereo range sensors. Each sensor produces a occupancy grid map separately. The final global occupancy map is an integration of both maps.

Obstacle avoidance using potential fields was introduced by Khatib (1986). The key idea of the potential fields was described as:

The manipulator moves in a field of forces. The position to be reached is an attractive pole for the end effector and obstacles are repulsive surfaces for the manipulator parts. Khatib (1986, p.91)

Any local minimum can cause the manipulator to stop at an unintended location where the local minimum exists. Volpe and Khosla (1987) presented superquadric artificial potentials to eliminate local minima. An obstacle is expressed in terms of elliptical isopotential, Fig. 2-6. The potential field method is conceptually simple; however, it has local minima that cannot be eliminated. It eliminates local minima only if one obstacle is associated with attractive goal potential; however, it still has local minima when many obstacles are cluttered in the workspace. Newman and Hogan (1987) also showed collision avoidance of moving obstacles via protective potential fields.

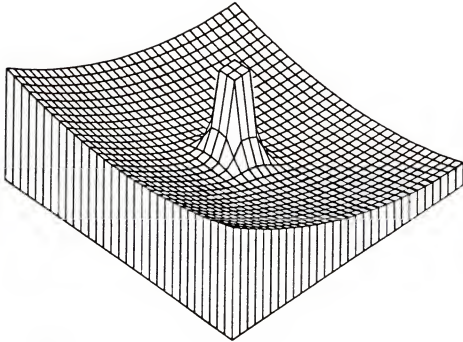
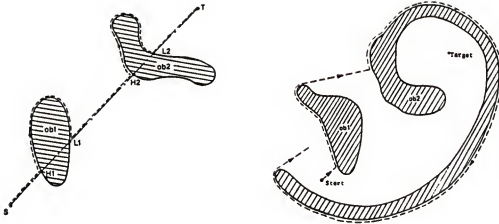


Fig. 2-6 A repulsive potential (for an obstacle) and an attractive well (for the target point), (from Volpe and Khosla, 1987)

Lumelsky (1985, 1987) and Lumelsky and Stepanov (1986) presented a path planning method resembling a bug's movement. A point follows a straight path from start position to goal position. When the point meets an obstacle, it then follows the left-side boundary of the obstacle until it returns to the straight path, Fig. 2-7,a. This algorithm does not find the shortest path nor the safest path, and may not guarantee termination in some situation, Fig. 2-7,b. A modification of Lumelsky's algorithm is found in Dupont and Derby (1988) in which a collision-free path is formed by searching from start position and goal position toward the mid point until they meet. The search strategy is the same as Lumelsky's.



(a) A path is found

(b) The method does not guarantee termination

Fig. 2-7 A path following the straight path and the boundary of obstacles, (from Lumelsky and Stepanov, 1986)

Obstacle avoidance as presented by Gilbert and Johnson (1985) is an algorithm that keeps away from obstacles by testing the distance between potentially colliding parts. Richbourg et al. (1987) introduced yet another method using Snell's law. Snell's law, an optic law, characterizes minimal cost paths in a routing problem. Given start and goal positions, the shortest path is found. The method calls for a time consuming optimization process. In addition, there exist blind regions that cannot be reached by any path obeying Snell's law.

Obstacle avoidance of a planar 3R robot was shown by Young and Duffy (1987) through articulation. The path was predefined. The allowable range of rotation of each link was computed at every step and the configuration of the robot was set to the mid-value of the allowable

range of rotation. Any obstacles that lie within the circular area set by the link reduce the allowable range of rotation. This in turn changes the configuration of the robot and results in articulation of the robot.

Maciejewski and Klein (1985) and Espiau and Boulic (1986) presented obstacle avoidance for redundant manipulators. In Maciejewski and Klein (1985), the method is to augment an obstacle avoidance Jacobian to the end effector Jacobian and then calculate joint angular velocity using a pseudo-inverse. The method actually relies on velocity control. In Espiau and Boulic (1986), proximity sensors are used to detect obstacles. The proximity information is incorporated in the Jacobian matrix as a repulsive force. The work presented by Hsu et al. (1988) was focused on the control of a redundant manipulator rather than obstacle avoidance. A dynamic control strategy (joint torque control) is used to guarantee the asymptotic tracking of a desired workspace trajectory.

Collision avoidance of multiple robots is another active research area as described by Freund and Hoyer (1985, 1986, 1988), Chien et al. (1988), Nagata et al. (1988) and Yeung and Bekey (1987). The common idea for collision avoidance of multiple robots is to set a different priority for each robot, and plan collision-free paths for each robot separately. Then the portions of paths that are in conflict with others are modified according to the priority, i.e., the lower priority path is modified.

2.2 Mobile Robots

Some of the major research efforts in mobile robots include the JPL robot by Thompson (1977), Stanford Cart by Moravec (1980), HILARE by Giralt et al. (1979, 1984), IMP (Intelligent Mobile Platform) by Crowley

(1985), ALV (Autonomous Land Vehicle) by Seida and Mogenthaler (1987) and HERMIES-IIB (Hostile Environment Robotic Machine Intelligence Experiment Series II-B) by Burks et al. (1987).

The JPL robot was one of the early experimental systems that was able to plan a path using a visibility graph where an obstacle was represented by algebraic curves. An obstacle was impassable, and the visibility graph was formed by connecting the edges of obstacles. The terrain model was provided by a vision system.

The Stanford Cart was also able to plan a path. Obstacles were represented by circles and the shortest path was found from the visibility graph.

HILARE at the LAAS laboratory in Toulouse, France, has been an experimental system since 1977. The path planning system used a connectivity graph which was a cellular decomposition of free space.

IMP requires a global model having a network of landmark points. A path planning algorithm provides a global path from the land marks.¹ A local navigation system carries out the planned global path, maintains an estimate of the robot's position with respect to the global model and plans local paths as needed to avoid unexpected obstacles.

ALV of Martin Marietta, Denver Aerospace Division, was capable of navigating a road. ALV was equipped with a color TV camera which captures video images for processing by the vision subsystem and a Laser Range Imager which captures an image of range values. The vision system locates road edges in the video image and obstacles in the range image. These

¹ A discussion of landmark recognition can be found in Nasr and Bhanu(1988).

images were used to generate a reference trajectory and for obstacle avoidance. An obstacle was approximated by a bounding box. The clearance between the road edge and the bounding box determined which side the vehicle would pass.

ALV was also used for testing a vision system developed by Daily et al. (1988) and Waxman et al. (1987). In Daily et al. (1988), the actual range image was a rectangular array that was used to build the Cartesian Elevation Map. An obstacle was represented by an elevated image on the grid. A path plan was carried out based on the map; a high level reflexive path planner set a series of subgoals which described the preferred route from start to goal. In Waxman et al. (1987), the scene analysis for navigation of ALV was focused. Road edges were represented by connection of straight line segments to provide long-, intermediate-, and short-range navigation. The paths generated were a "sequence of regions" (a long-range), a directional heading or "corridor of free space" connecting the current region with the next region (an intermediate-range), and a "track of safe passage" through the selected corridor while avoiding obstacles (short-range).

HERMIES-IIB was a sensor based mobile robot, developed at CESAR of Oak Ridge National Laboratory. HERMIES-IIB research had focused on navigation in unknown environments and visual perception. HERMIES-IIB was equipped with an IBM-7532 (an industrial version of the IBM-AT) with 2 NCUBE parallel processing boards for independent control and scene analysis.

The control system for a mobile robot requires the integration of sensors, modeling of the world, path planning and task execution. Chatila

(1986) proposed a hierarchical modular decomposition for mobile robot navigation. Iyengar et al. (1985) proposed a method of robot navigation which required no pre-learned model. As more information is consolidated into the terrain model, the building polygons of the obstacle fit more closely and the polygons representing free space grow larger. Brooks (1986b, 1987) proposed a layered control system for a mobile robot. The control system was decomposed into parallel modules based on task-achieving behavior which is contrary to the conventional ways of series decomposition of functional units. Ish-Shalom (1987) proposed the Funnel algorithm for task level robot control. The idea is to partition the task into a sequence of local sub-tasks and each sub-task is carried out by local feedback law.

2.3 Concluding Remarks

Most of the works related to collision avoidance and path planning are still in beginning stages. Algorithms used are either explicit descriptions of free space or the potential function method. Direct comparisons between algorithms are not possible because no standard problem settings, i.e. the size of workspace, the number, shape and size of obstacles, have been proposed. However the operations involved in these previous algorithms are known to be computationally expensive. Some of the notable simulations are those by Brooks (1983b), Brooks and Lozano-Pérez (1985), Espiau and Boulic (1986), Gouzènes (1984), Kambhampati and Davis (1986), Maciejewski and Klein (1985), Morevec (1980) and Volpe and Khosla (1987) and Young and Duffy (1987). Hopcroft, Schwartz and Sharir (1984) suggested a new research direction to the problem:

... even a certain simple two-dimensional case of the coordinated motion planning problem for multiple independent objects is PSPACE-hard¹. This suggests that the problem of planning coordinated motions of many bodies cannot be solved efficiently in the general case, a result that should direct further research either toward identifying special cases of the problem that are tractable or toward finding more practical approximation techniques. Hopcroft et al. (1984, p.87)

A new algorithm presented in the following chapters does not involve constructing free space or potential fields. The algorithm directly addresses collision-free paths in the form of polynomial curves.

¹ This indicates the difficulty of the problem in that the number of geometric operations is polynomial such as n^5 shown in p.11.

CHAPTER 3

CONCEPTUAL DEVELOPMENT OF COLLISION-FREE PATHS

In this chapter fundamentals of parametric curves that are essential to the development of collision-free paths are introduced. Here a collision-free path refers to a series of points connecting two designated points without interfering any obstacles. A comprehensive treatment on parametric curves and surfaces can be found in a book by Faux and Pratt (1979).

After the treatment of parametric curves, fundamental concepts essential for the generation of collision-free paths such as geometric mapping, selection of intermediate connection points are discussed. A Geometric mapping transforms obstacles in Euclidean space to parameter space. Images of obstacles in the parameter space of the mapping are deformed and may overlap each other. An important characteristic of a geometric mapping is that a point from the clear area in parameter space corresponds to a collision-free path in Euclidean space.

Intermediate connection points are required when the parameter space is completely occupied by images of obstacles. Selection of intermediate points is made near the vertices of obstacles that block the straight path between start and target points. The geometric mapping for each subinterval is then carried out to find a collision-free path for the subinterval.

A reduction of the three dimensional collision-free path planning to a two dimensional problem reduces the algebraic complexity of geometric mapping. The procedure involves projection of three dimensional obstacles onto two dimensions, thereby eliminating one variable. A projection plane is formed in such a way that it contains start and target points; thus, the two dimensional geometric mapping can be applied for the images on the projection plane.

3.1 Parametric Curves

There are several ways to describe curves, namely explicit form, implicit form, and parametric form. An explicit form of equation is expressed in a form such as $Y = f(X)$ for the two dimensional case. An inverse of the equation, $X = g(Y)$, may not be expressed in a closed form, and also there are curves that cannot be described in explicit form. An implicit form of equation is expressed as, for example, $f(X,Y) = 0$ for the two dimensional case, which has similar features of the explicit form.

The parametric form of equations have been studied mainly for applications of CAD/CAM and computer graphics. A parametric curve is expressed in the form, for example, $X = Q(s)$, $Y = R(s)$ for the two dimensional case where s is the parameter. Points on a parametric curve can be obtained sequentially along the curve by varying the parameter. There are many ways of representing parametric curves including spline and Bézier curves. Each representation of parametric curves has special characteristics; therefore, it is important to choose the proper one. A Bézier curve of order two is used throughout this dissertation. This particular curve greatly simplifies the geometric mapping in which the

mapping of obstacles in Euclidean space to images in parameter space as a closed form solution has been developed. This is explained in sections 3.2 and 3.3. Other curves will require intense computation to do such mapping by iteration.

3.1.1 The Bézier Curve

Any arbitrary continuous curve, $f(x)$, of finite length can be approximated to any desired degree of accuracy by a polynomial, $p(x)$, which is known as the Weierstrass theorem, Goult et al. (1973). A higher degree of accuracy will demand a higher degree for the polynomial $p(x)$. The polynomial $p(x)$ can be represented by a combination of some basis polynomial. One such approximation takes the following form:

$$p(x) = \sum_{k=0}^n \frac{n!}{k!(n-k)!} f(k/n) x^k (1-x)^{n-k} = \sum_{k=0}^n f(k/n) B_k(x) . \quad (3.1)$$

This is known as the Bernstein polynomial approximation of degree n . The basis polynomial is called the Bernstein basis for the set of all polynomials of degree n and is expressed as follows:

$$B_k(x) = \frac{n!}{k!(n-k)!} x^k (1-x)^{n-k} \quad (0 \leq x \leq 1.0) \quad (3.2)$$

where $k = 0, 1, 2, \dots, n$. Since the basis is defined in the interval $0 \leq x \leq 1$, in some cases it is necessary to transform an arbitrary interval, $a \leq \tau \leq b$, to the interval $0 \leq x \leq 1$. The desired finite interval $a \leq \tau \leq b$ can be transformed to $0 \leq x \leq 1$ by the formula $\tau = (b-a)x + a$. This linear relationship between x and τ does not distort the original curve. Topics related to polynomials are discussed at greater length, for example, in Goult et al. (1973).

Bézier (1972) extended the idea of approximation of a function to approximation of a polygon, in which $n+1$ vertices of a polygon are approximated via the Bernstein basis. If the scalars $f(k/n)$, a known function value at $x=k/n$, are replaced by the polygon vertices consecutively in terms of X, Y, Z coordinate values, the resulting vector-valued polynomial represents a Bézier curve segment of degree n . It is also called a Bernstein-Bézier polynomial curve. For instance the X coordinate of the Bézier curve is expressed as follows;

$$X(s) = \sum_{k=0}^n \frac{n!}{k!(n-k)!} r_{kX} s^k (1-s)^{n-k} \quad (3.3)$$

where r_{kX} is the X coordinate of vertex r_k , s is a parameter and the valid interval is $0 \leq s \leq 1$. Y and Z coordinates are expressed in a similar fashion. The parameter x in (3.1) is replaced by s and it should not be confused with the X coordinate of the vertices. When $n=2$, the basis consists of $B_0(s)=(1-s)^2$, $B_1(s)=2s(1-s)$ and $B_2(s)=s^2$, and the resulting curve is a parabola and it approximates the triangle:

$$\begin{aligned} \underline{r} &= \underline{r}_0(1-s)^2 + \underline{r}_1 2s(1-s) + \underline{r}_2 s^2 \\ &= (\underline{r}_0 - 2\underline{r}_1 + \underline{r}_2)s^2 + 2(\underline{r}_1 - \underline{r}_0)s + \underline{r}_0 \end{aligned} \quad (3.4)$$

The equation (3.4) can be derived geometrically, see Fig. 3-1. Points $\underline{r}_a$, $\underline{r}_b$ and $\underline{r}_c$ divide line segments $\underline{r}_0\underline{r}_1$, $\underline{r}_1\underline{r}_2$ and $\underline{r}_a\underline{r}_b$, respectively, by a ratio $s:(1-s)$. The parabolic curve is a path traced out by point $\underline{r}_c$, which is an approximation to the triangle $\underline{r}_0\underline{r}_1\underline{r}_2$. The curve can be derived as follows;

$$\begin{aligned} \underline{r}_a &= (\underline{r}_1 - \underline{r}_0)s + \underline{r}_0 \\ \underline{r}_b &= (\underline{r}_2 - \underline{r}_1)s + \underline{r}_1 \\ \underline{r}_c &= (\underline{r}_b - \underline{r}_a)s + \underline{r}_a \end{aligned}$$

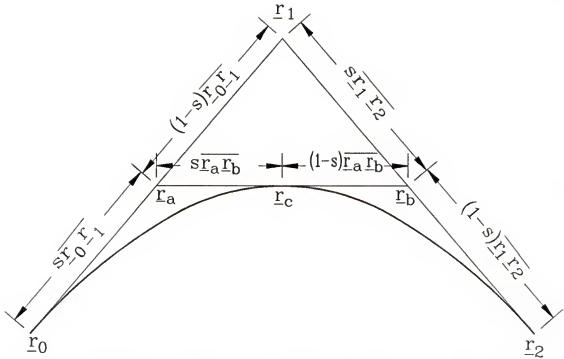


Fig. 3-1 Geometric construction of Bézier curve of order two

Substituting r_a and r_b in the right side yields

$$r_c = (r_0 - 2r_1 + r_2)s^2 + 2(r_1 - r_0)s + r_0$$

The valid interval of the parameter s is $0 \leq s \leq 1$ where $s=0$ corresponds to point r_0 and $s=1$ corresponds to point r_2 . The process can be generalized for a polygon of n vertices and a Bézier curve of order $n-1$ results. One should note that in Fig. 3-1 the lines r_0r_1 and r_1r_2 are tangents to the curve at points r_0 and r_2 respectively. The location of vertex r_1 determines the shape of the parabola. The parametric parabola derived above has special meaning in developing collision-free paths in which r_0 , r_2 and r_1 are viewed as start, target and control point, respectively.

3.1.2 Tangents and Normals to a Parametric Curve

Tangents and normals are important properties of a curve. Tangents and normals to a curve in explicit form are well known from algebraic

geometry. The following discussion shows the relationship between explicit form and parametric form. A tangent at point (x_0, y_0) on a curve $y=f(x)$ is expressed as follows:

$$y - y_0 = \left. \frac{dy}{dx} \right]_{(x_0, y_0)} (x - x_0) \quad . \quad (3.5)$$

It is possible to derive a tangent in parametric form from this equation with an assumption that the curve has a known parametric form $P(x(s), y(s))$ with s being the parameter. The equation is rearranged in terms of x and y using the chain rule. A substitution of $s=s_0$ for the point (x_0, y_0) yields

$$\left. \frac{ds}{dy} \right]_{s=s_0} (y - y_0) = \left. \frac{ds}{dx} \right]_{s=s_0} (x - x_0) = u \quad (3.6)$$

where u is a new parameter along the tangent. The tangent is

$$x(u) = x_0 + u \left. \frac{dx}{ds} \right]_{s=s_0} \quad (3.7)$$

$$y(u) = y_0 + u \left. \frac{dy}{ds} \right]_{s=s_0} \quad . \quad (3.8)$$

The normal of the curve at point $s=s_0$ can be derived in a similar fashion where the derivative of equation (3.5) is replaced by $-dx/dy$:

$$x = x_0 + u \left. \frac{dy}{ds} \right]_{s=s_0} \quad (3.9)$$

$$y = y_0 - u \left. \frac{dx}{ds} \right]_{s=s_0} \quad . \quad (3.10)$$

3.1.3 Coordinate and Parameter Transformation

Rotation or translation of the Bézier curve can be achieved by simply transforming the vertices that define the curve. A coordinate transformation of a parametric parabola is carried out by multiplying by transformation matrix M . The equation (3.4) can be written in a matrix

form in which $\underline{x}$, $\underline{x}_0$, $\underline{x}_1$ and $\underline{x}_2$ are column vectors in homogeneous coordinates:

$$\underline{x} = [\underline{x}_0 \ \underline{x}_1 \ \underline{x}_2] \begin{bmatrix} 1 & -2 & 1 \\ 0 & 2 & -2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ s \\ s^2 \end{bmatrix} = R U S \quad (3.11)$$

Multiplying by the transformation matrix yields

$$\begin{aligned} M \underline{x} &= M [\underline{x}_0 \ \underline{x}_1 \ \underline{x}_2] \begin{bmatrix} 1 & -2 & 1 \\ 0 & 2 & -2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ s \\ s^2 \end{bmatrix} = M R U S \\ &= R^* U S \end{aligned} \quad (3.12)$$

Equation (3.12) shows that a transformation of vertices, $R^* = M R$, is equivalent to the transformation of parametric curves.

The curve, however, may be constructed by a composite of different parametric curves each having different intervals. It may be required to make each curve have the same interval, such as from 0 to 1. A parameter transformation takes the interval $s_0 \leq s \leq s_1$ to the new interval $0 \leq s^* \leq 1$. The parameter transformation must be linear in order to preserve the degree of the polynomial. It is expressed in the following form, for the first order term:

$$s = (1-s^*)s_0 + s^*s_1 = s_0 + (s_1-s_0)s^* \quad (3.13)$$

A higher degree of s can be obtained from the formula

$$s^n = \sum_{i=0}^n \binom{n}{i} s_0^i ((s_1-s_0)s^*)^{n-i} \quad (3.14)$$

which is a binomial expansion. For instance, the column vector representing parameter s in equation (3.12) can be transformed from interval $s_0 \leq s \leq s_1$ to $0 \leq s^* \leq 1$ by substitution using equations (3.13) and (3.14):

$$\begin{bmatrix} 1 \\ s \\ s^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ s_0 & (s_1 - s_0) & 0 \\ s_0^2 & 2s_0(s_1 - s_0) & (s_1 - s_0)^2 \end{bmatrix} \begin{bmatrix} 1 \\ s^* \\ s^{*2} \end{bmatrix}$$

$$= T S^* \quad (3.15)$$

where T is the transformation matrix and the new parameter is defined in the interval $0 \leq s^* \leq 1$. T is nonsingular unless s_0 and s_1 are coincident. A parameter transformation can also be represented as a transformation of vertices:

$$\underline{x} = R U S = R U T S^* \quad (3.16)$$

A matrix R' representing the new vertices for $\underline{x}$ may be given by:

$$R' = R U T U^{-1} \quad (3.17)$$

then

$$R' U = R U T \quad (3.18)$$

Substituting (3.18) to (3.16) yields:

$$\underline{x} = R' U S^* \quad (3.19)$$

This is the same form of equation as (3.11), and the matrix U is always nonsingular, equation (3.11). The matrices T and T^* become identity matrices if the valid interval of s is $0 \leq s \leq 1$.

3.1.4 Smooth connection of two parametric curves

It is often required to connect two curves and to maintain continuity of the first derivative (c^1 continuity) or continuity of both

the first and second derivative(c^2 continuity) of the entire curve. In the following subsections, four different methods of smooth connection in c^1 continuity are described. Curves to be connected are denoted by C_0 and C_1 in the figures, and they are not necessarily touching each other. The curve indicated by C_2 is the smoothing curve.

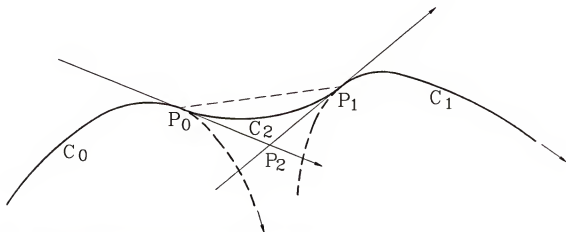


Fig. 3-2 Connection by using a parabola

Connection by a parabolic curve

Fig. 3-2 shows a smooth connection via a parabolic curve, a Bézier curve of order two. P_0 is a point on curve C_0 and P_1 is a point on curve C_1 . Tangents at points P_0 and P_1 are obtained by using equations (3.7) and (3.8), and P_2 is the intersection point of the tangents. P_0 , P_2 and P_1 form a triangle and a connecting parabola can be constructed using a Bézier curve. The solid line is the resulting smooth curve. This method cannot be used when two tangents are parallel or the intersection point is out of bounds, or a sharp edge at the connection point occurs as in Fig. 3-3. The smoothing process alters the valid interval of parameter of curve C_0 and C_1 . A parameter transformation may be needed to result in the same interval for each curve C_0 , C_1 and C_2 .

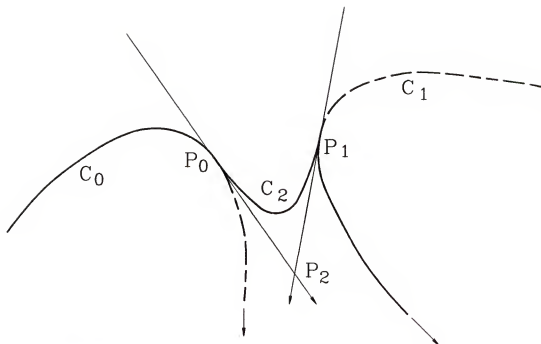


Fig. 3-3 Unsatisfactory connection

Direct combination of two parametric curves

Fig. 3-4 shows a combination of the specified interval of two curves. The combination can be viewed as a blending of two curves. The combination basis function can be devised to provide a c^1 or c^2 continuity at connection points. In this subsection a combination basis function that gives c^1 continuity is discussed. A linear combination of two curves cannot be used since it does not provide c^1 continuity at the connection point. The curves C_0 and C_1 have their own parameters so there are two different parameters associated with the curves. To reduce the number of independent parameters, the valid interval of each parameter is transformed to make the parameter interval the same. This allows the two different parameters to be treated as one parameter. The curves $P_1(s)$ and $P_{11}(t)$ in Fig. 3-4, indicated by dotted lines, are the sections to be

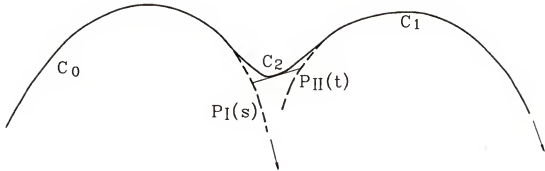


Fig. 3-4 Combination of two parametric curves

combined. The curve C_2 shows the combination of $P_I(s)$ and $P_{II}(t)$ and takes a form given by

$$C_2 = (1 - f(u))P_I(s) + f(u)P_{II}(t) \quad (3.20)$$

where $f(u)$ is the combination basis function in a parametric form. The range of parameter u can be set to be the same as s and t , $[0,1]$, so the parameters s , t and u are treated as one parameter. The combination basis function, $f(u)$, has to satisfy the following conditions to meet c^0 and c^1 continuity. A higher order continuity can be obtained by imposing additional conditions:

$$\begin{aligned} C_2(0) &= [P_I(s)]_{s=0} \\ C_2'(0) &= [P_I'(s)]_{s=0} \\ C_2(1) &= [P_{II}(t)]_{t=1} \\ C_2'(1) &= [P_{II}'(t)]_{t=1} \end{aligned} \quad (3.21)$$

These conditions require the combination basis function to be:

$$\begin{aligned} f(0) &= 0 \\ f(1) &= 1 \\ f'(0) &= 0 \\ f'(1) &= 0 \end{aligned} \quad (3.22)$$

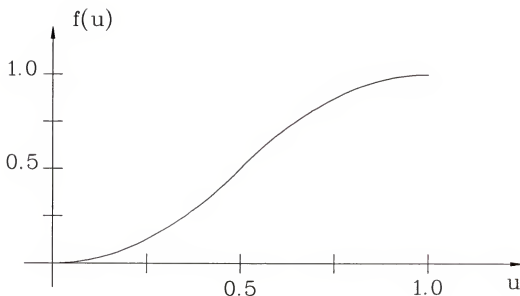


Fig. 3-5 Combination basis function

Equation (3.22) leads to a combination basis function of order three, a cubic function:

$$f(u) = u^2(3-2u) \quad (3.23)$$

A combination basis function based on equation (3.23) is shown in Fig. 3-5.

It is important to see that the valid interval of parameters of $P_I(s)$, $P_{II}(t)$ and $f(u)$ are the same, so they can be treated as one parameter.

Equation (3.20) yields

$$C_2(v) = (P_{II}(v) - P_I(v))v^2(3-2v) + P_I(v) \quad (3.24)$$

where v is the new parameter representing the parameters s , t and u . The order of the curve $C_2(v)$ is increased by the order of the combination basis function. The curve $C_2(v)$ is confined in a region surrounded by the curves $P_I(s)$ and $P_{II}(t)$.

Connection by cubic curve

Instead of blending sections of two curves, smoothing can be carried out using a cubic curve. Since four coefficients are involved in a cubic function, four conditions are necessary to establish the unknown coefficients. The conditions are c^0 and c^1 continuity at the connection points P_0 and P_1 . Fig. 3-6 shows the result of connection where $C_2(u)$ is the connecting cubic curve. $C_2(u)$ in parametric form is expressed in general by

$$C_2(u) = \begin{cases} x_2(u) = a_0 u^3 + b_0 u^2 + c_0 u + d_0 \\ y_2(u) = a_1 u^3 + b_1 u^2 + c_1 u + d_1 \end{cases} \quad (3.25)$$

where $a_0, a_1, b_0, b_1, c_0, c_1, d_0$ and d_1 are unknown coefficients. The valid interval of parameter u is $0 \leq u \leq 1$. The conditions are

$$\begin{aligned} C_0(s) \big|_{s=s^*} &= C_2(u) \big|_{u=0} \\ C_0(s)' \big|_{s=s^*} &= C_2(u)' \big|_{u=0} \\ C_1(t) \big|_{t=t^*} &= C_2(u) \big|_{u=1} \\ C_1(t)' \big|_{t=t^*} &= C_2(u)' \big|_{u=1} \end{aligned} \quad (3.26)$$

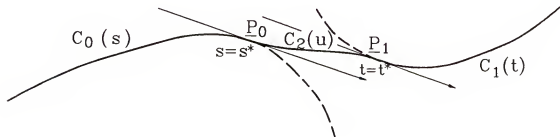


Fig. 3-6. Connection by a cubic curve

Substituting equation (3.26) into (3.25) with x and y components of $C_0(s)$ and $C_1(t)$ being $(x_0(s), y_0(s))$ and $(x_1(t), y_1(t))$ respectively, the unknown coefficients of equation (3.25) are obtained as follows:

$$d_0 = x_0(s^*) \quad (3.27)$$

$$d_1 = y_0(s^*) \quad (3.28)$$

$$c_0 = \left. \frac{dx_0(s)}{ds} \right]_{s=s^*} \quad (3.29)$$

$$c_1 = \left. \frac{dy_0(s)}{ds} \right]_{s=s^*} \quad (3.30)$$

$$a_0 = -2(x_1(t^*) - c_x - d_x) + \left. \frac{dx_1(t)}{dt} \right]_{t=t^*} - c_x \quad (3.31)$$

$$a_1 = -2(y_1(t^*) - c_y - d_y) + \left. \frac{dy_1(t)}{dt} \right]_{t=t^*} - c_y \quad (3.32)$$

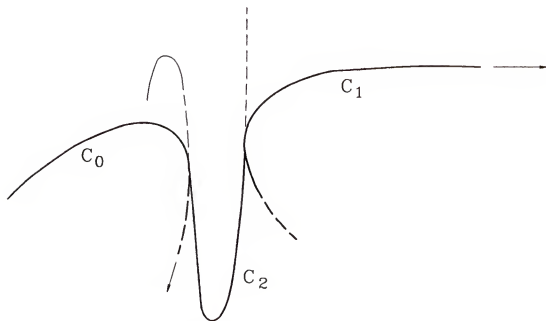


Fig. 3-7 Out of bound connection curve

$$b_0 = 3(x_1(t^*) - c_x - d_x) - \frac{dx_1(t)}{dt} \Big|_{t=t^*} + c_x \quad (3.33)$$

$$b_1 = 3(y_1(t^*) - c_y - d_y) - \frac{dy_1(t)}{dt} \Big|_{t=t^*} + c_y \quad (3.34)$$

This method requires calculation of derivatives. There is a possibility that the connecting cubic curve can be out of bounds, as shown in Fig. 3-7. These erratic properties of cubic connection curves have to be carefully considered before applying the method.

Connection by a quartic curve

If an additional point is specified for the connection curve to pass through, a quartic curve has to be used as a connection curve. The additional point can effectively limit the connection curve in certain regions. If a connection curve has to meet the requirements of c^0 and c^1 continuity at connection points as well as to pass through an additional point, a quartic curve connection is necessary because the additional point introduces another unknown variable. The additional point also can be used to control and limit the shape of the connection curve. Fig. 3-8 shows a quartic curve connection, where C_2 is a quartic connection curve and P_2 is the additional point. C_2 in parametric form is expressed as follows:

$$C_2(u) = \begin{cases} x_2(u) = a_0 u^4 + b_0 u^3 + c_0 u^2 + d_0 u + e_0 \\ y_2(u) = a_1 u^4 + b_1 u^3 + c_1 u^2 + d_1 u + e_1 \end{cases} \quad (3.35)$$

where a_0 , a_1 , b_0 , b_1 , c_0 , c_1 , d_0 , d_1 , e_0 and e_1 are unknown coefficients and the valid interval of u is $0 \leq u \leq 1$. The conditions are as follows:

$$C_0(s)]_{s=s^*} = C_2(u)]_{u=0} = P_0(P_{0x}, P_{0y})$$

$$C_0(s)']_{s=s^*} = C_2(u)']_{u=0} = P_0'(P_{0x}', P_{0y}')$$

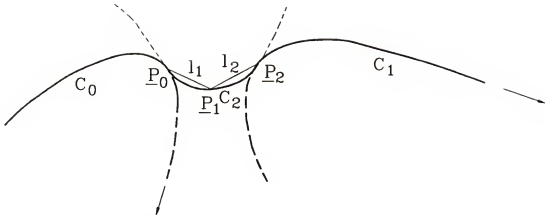


Fig. 3-8 Connection by a quartic curve

$$C_1(t)|_{t=t^*} = C_2(u)|_{u=1} = \underline{P}_1(P_{1x}, P_{1y}) \quad (3.26)$$

$$C_1(t)'|_{t=t^*} = C_2(u)'|_{u=1} = \underline{P}_1'(P_{1x}', P_{1y}')$$

$$C_2(u)|_{u=u^*} = \underline{P}_2(P_{2x}, P_{2y})$$

The coefficients d_0 , d_1 , e_0 and e_1 are easily calculated from the first two conditions of equation (3.35).

$$\begin{aligned} d_0 &= P_{0x} \quad , \quad d_1 = P_{0x}' \quad , \\ e_0 &= P_{0y} \quad , \quad e_1 = P_{0y}' \quad . \end{aligned} \quad (3.36)$$

The parameter indicating point $\underline{P}_1$ is u^* and it approximately can be set by the ratio of $\overline{P_0 P_1}$ to $\overline{P_1 P_2}$.

$$u^* = \frac{\overline{P_0 P_2}}{\overline{P_0 P_1} + \overline{P_1 P_2}} \quad (3.37)$$

By substituting equation (3.35) into (3.34), one can get the following linear equations for unknown coefficients a_0 , b_0 and c_0 :

$$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 3 & 4 \\ u^{*2} & u^{*3} & u^{*4} \end{bmatrix} \begin{bmatrix} a_0 \\ b_0 \\ c_0 \end{bmatrix} = \begin{bmatrix} P_{1x} - P_{0x} - P_{0x}' \\ P_{1x}' - P_{0x}' \\ P_{2x} - P_{0x} - P_{0x}' u^* \end{bmatrix} \quad (3.38)$$

and for a_1 , b_1 and c_1 :

$$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 3 & 4 \\ u^{*2} & u^{*3} & u^{*4} \end{bmatrix} \begin{bmatrix} a_1 \\ b_1 \\ c_1 \end{bmatrix} = \begin{bmatrix} P_{1y} - P_{0y} - P_{0y}' \\ P_{1y}' - P_{0y}' \\ P_{2y} - P_{0y} - P_{0y}'u^* \end{bmatrix} \quad (3.39)$$

It should be noted that the parameter indicating P_1 , u^* , is in fact arbitrary. It determines the overall shape of the connection curve, therefore, the proper value is important. The value given by equation (3.37) is an approximation to a parabolic curve.

3.2 Collision-Free Paths Connecting Two Designated Points in an Obstacle Strewn Two Dimensional Environment

In this section, the main ideas of planning collision-free paths are discussed. A collision-free path refers to a series of points connecting two designated points which does not interfere with any obstacles.

In this problem, the start and target points are given. Further, all obstacles are stationary, eliminating problems associated with moving obstacles. Obstacle location, size and shape are given by the coordinates of vertices.

This dissertation considers a collision-free path to be a polynomial curve connecting two designated points without intersecting any obstacles. In the previous chapter, several methods of planning collision-free path such as connectivity graphs, potential function method, free space search method and Bug's algorithm are reviewed. Here for the first time a collision-free path generation algorithm employing a Bézier curve of order two is introduced and simulated. The motivation

for use of a Bézier curve of order two is that it gives control of the shape of the curve, and permits a closed form solution for the geometric mapping. The geometric mapping transforms obstacles in Euclidean space to parameter space (θ, t) . The main property of geometric mapping is that a point in areas of parameter space which is not occupied by obstacle images corresponds to a collision-free path in Euclidean space. Hence the geometric mapping provides multiple choice of paths. Generalization of the geometric mapping based on a higher order Bézier curve is theoretically possible, however, an enormous amount of computation will be required to carry out such a geometric mapping as the solution will be approximated. The process of geometric mapping and selection of intermediate connection points are briefly explained later in this chapter and discussed in detail in Chapter 4.

The search for a collision-free path proceeds follows: a straight path joining the start and target point is examined. If the straight path is suitable then the problem becomes trivial. However, if it is blocked by an obstacle, a Bézier curve of order two is used to construct a collision-free path. If no such curve exists which connects the start and target points without intersecting any obstacle, a curve with higher degree of order is necessary to construct a collision-free path. As the geometric mapping of such higher order curves is not feasible for a real-time solution, the path is partitioned and a recursive application of parabolic curves for each subinterval of the start and target point is employed. Selection of intermediate connection points, geometric mapping of subintervals and smoothing the slope discontinuity at the connection

points are then carried out. An overall flow chart is given in section 4.5.

For extreme cases where the start or target point is surrounded by concave obstacles or within a maze and the other is outside, the algorithm may fail to find a path. This issue is discussed at the end of section 3.2.2.

3.2.1 Geometric Mapping

A Bézier curve of order two, a parametric parabola, is defined by three points as explained in Section 3.1.1. Here the equation (3.4) is restated in x and y components assuming that the curve is in the x-y plane. The curve, shown in Fig. 3.1, is defined by

$$\begin{aligned} x &= (1-s)^2 r_{0x} + 2s(1-s)r_{1x} + s^2 r_{2x} \\ y &= (1-s)^2 r_{0y} + 2s(1-s)r_{1y} + s^2 r_{2y} \end{aligned} \quad (3.40)$$

where s is the parameter and r_0 , r_1 and r_2 are the defining vertices. The end points r_0 and r_2 can be regarded as the start and target points, respectively. The shape of the parabola is determined by r_1 which is hereafter called the control point. The control point can be arranged in polar coordinates or in cartesian coordinates. In the following discussion polar coordinates are chosen over cartesian coordinate since better shape consistency between parameter values is obtained. By organizing the position of the control point in polar coordinates (θ, t) , intersections between parabolas and obstacles can be plotted in parameter space in terms of the value of (θ, t) .

In Fig. 3-9, the start point, S, and the target point, T, are equivalent to r_0 and r_2 , respectively, in equation (3.40). The control point, Q, which is equivalent to r_1 , is expressed in polar coordinates

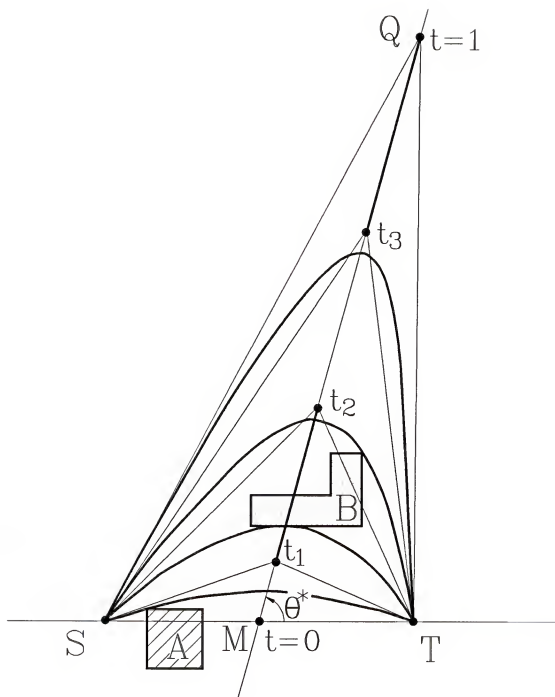


Fig. 3-9 Obstacles in Euclidean space

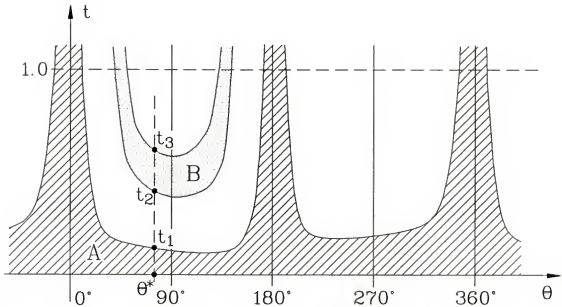


Fig. 3-10 Obstacle images in θ - t space

defined by the two parameters θ and t . The parameter θ determines a family of curves and the parameter t determines the length of the curves. By varying θ and t , every possible parabola that is inside the workspace and passes through S and T can be generated. Thus any obstacle that blocks parabolas creates an image on the θ - t space as shown in Fig. 3-10. The geometric mapping is defined to be a transformation of obstacles in Euclidean space to parameter space. Dimensions of a parameter space are determined by the total number of independent variables associated with control points. For instance, in Fig. 3-10, θ and t are the independent variables associated with the control point Q, hence two dimensional parameter space results.

When a single parabola intersects multiple obstacles their images in parameter space will overlap. Clear areas on a parameter space indicate collision-free parabolas. Figures 3-9 and 3-10 show this

relationship. In Fig. 3-9, for angle $\theta = \theta^*$, parabolas defined by t in the range $0 \leq t \leq t_1$ intersect obstacle A and those with t in the range $t_2 \leq t \leq t_3$ intersect obstacle B. The images in parameter space are mapped as shown in Fig. 3-10. A point having the smallest value of t in the clear area of parameter space corresponds to the shortest parabolic path between S and T. A path that may be regarded as the safest path can be picked from the clear area by finding the point that is farthest away from the surrounding obstacle images. The operations involved in obtaining obstacle images and selecting a point from parameter space are given in Chapter 4.

If obstacles are cluttered in the workspace, the parameter space will be completely filled with obstacle images. This indicates that no collision-free parabola is available. One solution is to use higher-order curves such as Bézier curves of order three. The main problem of using higher order curves is that the curve can not be controlled in the way of a parabolic curve. For Bézier curves of order three, four points are involved in defining the curve and two mid-points can be regarded as control points. A four-dimensional parameter space results because two parameters are associated with each control point. The obstacle images in such space are not as easily obtained as in two dimensional parameter space; also a higher order curve may contain loops in the designated interval. In the following section a practical alternate solution for these problems is addressed.

3.2.2 Recursion of Parametric Parabolas

As discussed in the previous section, a parameter space of dimension greater than two is difficult to manage. Instead of using

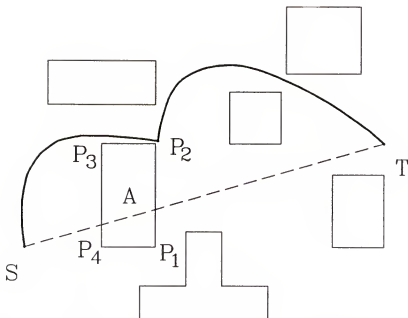


Fig. 3-11 Selection of an intermediate connection point

Bézier curves of order higher than two, a piece-wise connection of parabolic curves can be used. This approach reduces the original problem to a problem of determining the intermediate connection points. The geometric mapping can then be carried out for each subinterval to find collision-free path.

A connection point cannot be selected arbitrarily. It should be within the workspace and outside of obstacles. It is obvious that the obstacles that block the straight path should be circumvented. These obstacles can be identified by checking the intersection with the straight path ST. Points near the outer edge of these obstacles can be selected as intermediate connection points. For instance, in Fig. 3-11 vertices P₁ and P₄ are the right-side vertices, and P₂ and P₃ are the left-side vertices with respect to the line segment ST. An intermediate connection point can be selected in the region outside of obstacle A. It is

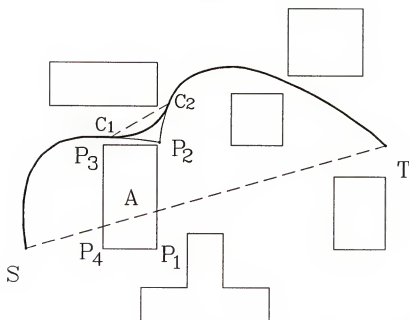


Fig. 3-12 Smoothing the slope discontinuity of two connected parabolas important to check whether the selected point is also outside of other obstacles to verify proper selection.

If the geometric mapping on a subinterval cannot provide a collision-free path, then that interval is further divided into smaller sections. In other words the process of finding collision-free paths is carried out in a recursive fashion.

A path so constructed has slope discontinuity at the connection points, and may be unsuitable for guiding a robot, see Fig. 3-11. Also the connection region is near the obstacle, as shown in Fig. 3-11, hence the path may be unsafe for guiding a robot. By making a smooth connection, the final path can be made suitable for guiding a robot further away from any obstacles, which makes a safer path, as shown in Fig. 3-12. There are several ways to achieve a smooth connection as described in Section 3.1.3. Fig. 3-12 shows a smooth connection of two

adjacent curves using the method of direct combination of two parametric curves. Since the smoothing process alters the original path, there is a chance that the connection curve may interfere with obstacles. The triangle $P_2C_1C_2$ in Fig.3-12 can be used to check for intersections of C_1C_2 with an obstacle. Thus the smoothing region can be adjusted to avoid obstacle interferences.

For some extreme cases, such as either the start or target point being completely surrounded by concave obstacles, or either point inside a maze, the entire parameter space is then occupied by obstacle images. Shown in Fig. 3-13 is a case where the start point is surrounded by concave obstacles and the target point is inside a maze. Obviously the parabolas connecting the points S, T will interfere with obstacles, and

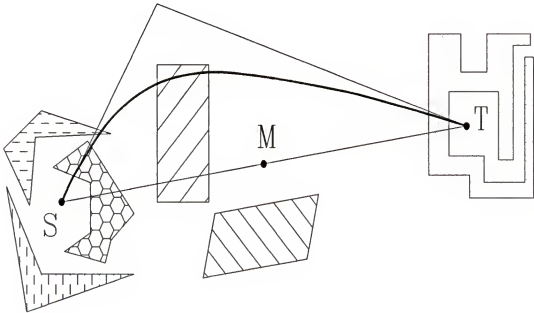


Fig. 3-13 Failure of planning collision-free path

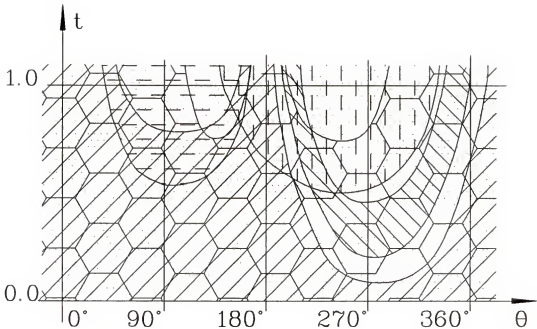


Fig. 3-14 Parameter space for Fig.5-13

the entire parameter space will be occupied by obstacle images, as shown in Fig. 3-14. After finding intermediate connection points, the geometric mapping for each subinterval is performed. If the entire parameter space for a subinterval is still occupied by obstacle images like Fig. 3-14, then algorithm will fail to generate a collision-free path for such cases. The same hatch patterns are used for obstacles in Fig. 3-13 and for the images of corresponding obstacles in Fig. 3-14.

3.3 Collision-Free Paths Connecting Two Designated Points in an Obstacle Strewn Three Dimensional Environment

In three dimensions, a control point can be expressed in spherical coordinates (ϕ, θ, t) as an extension of polar coordinates (θ, t) of two dimensions. If two control points are involved, parameter space becomes

six dimensional and geometric mapping is practically unfeasible. Methods developed for the two dimensional case are applicable to the three dimensional case. Direct manipulation of the three dimensional problem is, however, algebraically more complicated. This can be avoided by reducing the three dimensional problem to a two dimensional problem.

The following discussions illustrate projections of three dimensional objects onto a surface. The algorithm for two dimensional collision-free paths can then be used without modification. Thus planning collision-free paths in three dimensions is decomposed into two steps: projection of obstacles into a plane and determination of valid paths.

Fig. 3-15 shows a three dimensional obstacle and a Bézier curve of order two which contains one control point denoted by P. The three parameters ϕ , θ and t determine the position of P. The parameter ϕ merely determines the inclination angle of the flat plane, which contains the curve itself, from the reference plane set by start point, S, target point, T, and the origin of the workspace, O. The reference plane can be a vertical plane containing the points S and T. Thus the geometric mapping of the obstacle images on the flat plane can be obtained for the fixed value of ϕ in the range $0 \leq \phi \leq \pi$. In other words, the flat plane is a projection plane and obstacle images on the plane can be treated as two dimensional obstacles so the geometric mapping developed previously can be applied. A three dimensional geometric mapping results by stacking the two dimensional geometric mapping for varying values of ϕ . A collision-free path is searched on the projection plane for varying the parameter values from $\phi=0^\circ$ to $\phi=180^\circ$. After varying the parameter values from $\phi=0^\circ$ to $\phi=180^\circ$, the search may not find a collision-free path if all of the

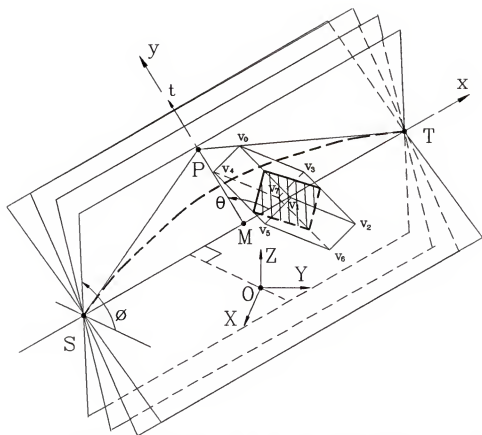


Fig. 3-15 Intersection of an obstacle with flat projection planes

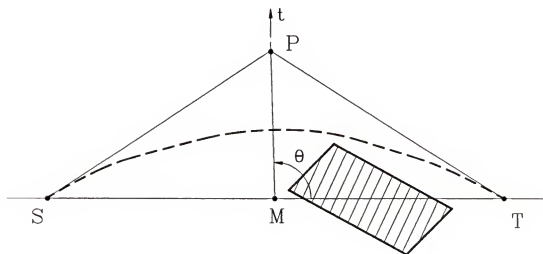


Fig. 3-16 Image of obstacle on the projection plane defined by $\emptyset$

projection planes do not produce a collision-free path. This indicates a failure of finding a collision-free path. Other search methods for three dimensional case are not attempted.

Fig. 3-16 shows the image of the obstacle given in Fig. 3-15, which is the intersection of the obstacle with the projection plane. The vertices of the obstacle image are the intersection points of edges of the obstacle with the projection plane. The projection plane is defined by three points S, T, and P where P determines the inclination of the plane. The plane in implicit form can be expressed as follows:

$$Ax + By + Cz + D = 0 \quad . \quad (3.41)$$

An obstacle edge is defined by two end vertices and can be described in a parametric form such as:

$$\begin{aligned} x &= (V_{1x} - V_{0x})t + V_{0x} \\ y &= (V_{1y} - V_{0y})t + V_{0y} \end{aligned} \quad (3.42)$$

$z = (V_{1z} - V_{0z})t + V_{0z}$
 where t is the parameter and the valid interval of t for the edge is $0 \leq t \leq 1$. Intersection of an edge of the obstacle with the projection plane can be determined by substituting equation (3.42) into (3.41). If the solution for t is in $0 \leq t \leq 1$, intersection occurs and the intersection point is given by equation (3.42). Otherwise, no intersection occurs. Intersection points so calculated have to be organized as a closed polygon as in Fig. 3-16. Images on the projection plane are expressed in terms of three dimensional coordinates. It is, therefore, necessary to transform them to two dimensional coordinates in order that the algorithm developed for two dimensional collision-free paths can be used without modification. The two dimensional collision-free path is then transformed

back to three dimensions. This is a coordinate transformation which was explained previously in Section 3.1.3.

3.4 Concluding Remarks

This chapter has described the essential concepts of the geometric mapping, recursion of parabolas and reduction of three dimensional problems to two dimensional problems. The geometric mapping transformed obstacles in Euclidean space to a parameter space in (θ, t) in which a Bézier curve was the vehicle for the mapping. When Bézier curves of order two cannot provide a collision-free path, recursion of parabola is used rather than geometric mapping based on higher order Bézier curves of order higher than two. Projection onto a flat plane was introduced to reduce the complexity of the three dimensional problem.

In summary, a collision-free path consists of parabolas. Since curvature of a parabola cannot be controlled easily, a collision-free path may be unnecessarily large. This problem can be alleviated by finding the shortest path which corresponds to the smallest value of t in parameter space Figures 3-9 and 3-10.

The process of finding collision-free paths breaks down, however, if the start or target point is inside of a maze, or if it is partially or totally enclosed by multiple obstacles or by concave obstacles as shown in Figures 3-13 and 3-14. It is also possible that an intermediate connection point can be partially enclosed by overlapping obstacles or concave obstacles, and as a result the process may not proceed further. In implementing the algorithm, such events can be monitored by noting the depth of recursion.

CHAPTER 4

ANALYSIS OF COLLISION-FREE PATHS

In this chapter, detailed algorithms for planning of a collision-free path are presented. Recall that a collision-free path is viewed as a polynomial-curve, connecting user specified start and target points, without intersecting any stationary obstacles. First the straight paths connecting start and target point for two and three dimensional cases are examined before carrying out the geometric mapping to determine whether such a path is feasible. The geometric mapping consists of the formulation of a parametric parabola, interference of the parabola with obstacles and selection of desired collision-free path from parameter space. This is followed by recursion of parametric paths which includes the selection of intermediate connection points, fundamental mode of paths and smoothing the sharp connection. In three dimensional environments collision-free paths are generated by first projecting three dimensional obstacles onto two dimensions to eliminate one variable of the geometric mapping.

4.1 Straight Paths

4.1.1 Straight Path in Two Dimensions

Fig. 4-1 shows a two dimensional robot work environment where start and target point are denoted by $S=(S_x, S_y)$, $T=(T_x, T_y)$, respectively, and

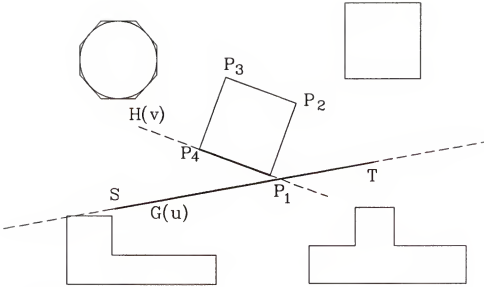


Fig. 4-1 Straight path and two dimensional obstacles

the circular obstacle is simplified by an octagon. The straight path from start to target point is represented by a parametric equation, $G(u)$, where u is the parameter, and indicated by a solid line segment ST . The validity of the straight path is examined by checking for intersections of the line segment ST with edges of obstacles. Edges of obstacles also are represented by a parametric equation, for instance the edge P_4P_1 is given by $H(v)$ where v is the parameter.

$$G(u) \begin{cases} x(u) = (T_x - S_x)u + S_x \\ y(u) = (T_y - S_y)u + S_y \end{cases} \quad (4.1)$$

$$H(v) \begin{cases} x(v) = (P_{1x} - P_{4x})v + P_{4x} \\ y(v) = (P_{1y} - P_{4y})v + P_{4y} \end{cases} \quad (4.2)$$

At intersection, $G(u) = H(v)$ and the solutions of u and v are as follows;

$$u = \frac{(P_{4y} - S_y)(T_x - S_x) - (P_{4x} - S_x)(T_y - S_y)}{(P_{1x} - P_{4x})(T_y - S_y) - (P_{1y} - P_{4y})(T_x - S_x)} \quad (4.3)$$

$$v = \frac{(P_{1y} - P_{4y})(S_x - P_{4x}) - (P_{1x} - P_{4x})(S_y - P_{4y})}{(P_{1x} - P_{4x})(T_y - S_y) - (P_{1y} - P_{4y})(T_x - S_x)} \quad (4.4)$$

When the solution of v from (4.4) is $0 \leq v \leq 1$, the edge of the obstacle does intersect with the line ST , however it does not necessarily intersect with the line segment ST . In addition, if the solution of u from (4.3) is $0 \leq u \leq 1$, the edge of the obstacle intersects with the line segment ST . If the denominator of (4.2) vanishes, edge P_4P_1 and line ST are parallel or collinear; such special cases can be considered as non-intersecting. The calculation has to be carried out for all edges of the obstacles to determine the interference between the obstacles and line segment ST . If all edges of an obstacle do not intersect with line ST , the obstacle is located on one side of line ST . The location of the obstacles can be determined by observing the sign of determinant, Δ , formed with S , T and a vertex of the obstacle:

$$\begin{vmatrix} 1 & 1 & 1 \\ S_x & T_x & P_{0x} \\ S_y & T_y & P_{0y} \end{vmatrix} \quad (4.5)$$

If the sign of the determinant is positive, the obstacle is above the line ST , otherwise it is below the line ST .

4.1.2 Straight Path in Three Dimensions

Interference of the straight path with obstacles can be identified in a way similar to planar case. The difference is that intersections are between surfaces of obstacles and the straight path. In Fig. 4.2, the line $L(u)$ representing line segment ST is given by

$$L(u) = (T - S)u + S \quad (4.6)$$

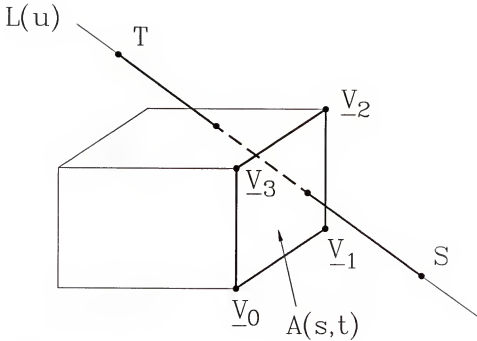


Fig. 4-2 Straight path and three dimensional obstacle

where the parameter u is defined to be $0 \leq u \leq 1$ for the line segment ST . The surface $A(s,t)$ defined by vertices V_0 , V_1 , V_2 and V_3 can be expressed in a bilinear parametric equation, where s and t are parameters,

$$A(s,t) = \begin{bmatrix} 1-s & s \end{bmatrix} \begin{bmatrix} V_0 & V_3 \\ V_1 & V_2 \end{bmatrix} \begin{bmatrix} 1-t \\ t \end{bmatrix} \quad (4.7)$$

where the parameter range of s and t are $0 \leq s \leq 1$ and $0 \leq t \leq 1$. The condition for finding the intersection point of the surface $A(s,t)$ and the line $L(u)$ is

$$A(s,t) = L(u) \quad (4.8)$$

After substituting equations (4.6) and (4.7) into (4.8), and rearranging the equation yields:

$$A_x s t + B_x s + C_x t + D_x u + E_x = 0$$

$$A_y s t + B_y s + C_y t + D_y u + E_y = 0 \quad (4.9)$$

$$A_z s t + B_z s + C_z t + D_z u + E_z = 0$$

$$\text{where } A_x = V_{0x} - V_{1x} + V_{2x} - V_{3x}$$

$$B_x = -V_{0x} + V_{1x}$$

$$C_x = -V_{0x} + V_{3x}$$

$$D_x = S_x - T_x$$

$$E_x = V_{0x} - S_x$$

$A_y, A_z, B_y, B_z, C_y, C_z, D_y, D_z, E_y$ and E_z are calculated similarly by substituting y and z components of S, T, V_0, V_1, V_2 and V_3 . Eliminating u from equation (4.9) results in equations in s and t .

$$a_x s t + b_x s + c_x t + d_x = 0$$

$$a_y s t + b_y s + c_y t + d_y = 0 \quad (4.10)$$

$$\text{where } a_x = A_x D_z - A_z D_x \quad a_y = A_y D_z - A_z D_y$$

$$b_x = B_x D_z - B_z D_x \quad b_y = B_y D_z - B_z D_y$$

$$c_x = C_x D_z - C_z D_x \quad c_y = C_y D_z - C_z D_y$$

$$d_x = E_x D_z - E_z D_x \quad d_y = E_y D_z - E_z D_y$$

Again s or t can be eliminated from (4.10) to obtain a quadratic equation in t or s . Elimination of t leads to a quadratic equation in s :

$$(a_y b_x - a_x b_y) s^2 + (b_x c_y - c_x b_y + a_y d_x - a_x d_y) s + (c_y d_x - c_x d_y) = 0 \quad (4.11)$$

Since there is only one intersection between a line and a plane surface, the discriminant of (4.11) will vanish and the solution of s has double roots. The double roots of s are given by

$$s = \frac{b_x c_y - c_x b_y + a_y d_x - a_x d_y}{2 (a_x b_y - a_y b_x)} \quad (4.12)$$

and a corresponding solution for t can be obtained from one of the equations of (4.10)

$$t = - \frac{b_y s + d_y}{a_y s + c_y} \quad (4.13)$$

The solution for u can be obtained by substituting the solutions of s and t into one of the equations of (4.9) provided that the denominator does not vanish. If the solutions for s , t and u are in the ranges $0 \leq s \leq 1$, $0 \leq t \leq 1$ and $0 \leq u \leq 1$ an intersection occurs, since $0 \leq s \leq 1$ and $0 \leq t \leq 1$ represents the surface patch defined by vertices V_0 , V_1 , V_2 and V_3 of the obstacle and $0 \leq u \leq 1$ represents line segment ST .

If the denominator of equation (4.12) or (4.13) vanishes, the surface and the line are parallel or coplanar and such case can be considered as non-intersecting. The calculation has to be carried out for all sides of the obstacles to determine interference of obstacles with the straight path ST .

One may note that the algorithm for checking interference given above can be used to check for collision of a robot with obstacles. Line segments representing robot links are substituted for line segment ST in equations (4.1) through (4.9). This feature is used for collision prevention of a robot discussed in Chapter 5.

4.2 Geometric Mapping of Two Dimensional Obstacles

4.2.1 Formulation of a Parametric Parabola

In this section, the formulation of a parametric parabola including the method of setting the ranges of parameters is explained. Fig. 4.3 shows a hypothetical workspace specified by a circle of radius r and a

step before obtaining $L_q(t)$ which is defined by M and Q. The workspace has to be referenced in parameter space before carrying out the geometric mapping of obstacles. The range of θ spans from 0° to 360° and the range of t , yet to be set, varies from 0 to 1. Thus a circular workspace in Euclidean space is mapped into a rectangular area in parameter space.

This section addresses the method of finding the point Q which defines the largest parabola bounded in workspace. When parabola coincides with line ST, t is set to zero, and the corresponding control point coincides with point M. When parabola is in contact with the boundary of workspace, t is set to one, and the corresponding control point is Q. The point R' , R and Q have to be obtained from the given points S, T and C, and the radius of workspace r , and an inclination angle θ . The midpoint of S and T is denoted by M, and is the center of rotation of line L_q . The point M is specified by

$$M = (S + T)/2 \quad . \quad (4.14)$$

The point R' is a rotation of T about M by angle θ :

$$\begin{cases} R'_x = M_x + (T_x - M_x)\cos\theta \\ R'_y = M_y + (T_y - M_y)\sin\theta \end{cases} \quad . \quad (4.15)$$

The line $L_q(t')$ defined by points M and R' can now be obtained. The line $L_q(t')$, in parametric form, is

$$\begin{cases} x = (R'_x - M_x)t' + M_x \\ y = (R'_y - M_y)t' + M_y \end{cases} \quad . \quad (4.16)$$

The circle centered at C with radius r is expressed as follows:

$$(x - C_x)^2 + (y - C_y)^2 = r^2 \quad . \quad (4.17)$$

The point R is the intersection of the circle and the line $L_q(t')$; the solution can be obtained in terms of parameter t' in which t' should be

greater than zero. By substituting (4.16) to (4.17) a quadratic equation in t' is obtained, and the solution of t' is

$$t' = \frac{-(ac+bd) + ((ac+bd)^2 - (a^2+b^2)(c^2+d^2-r^2))^{1/2}}{a^2 + b^2} \quad (4.18)$$

where

$$\begin{aligned} a &= R_x' - M_x \\ b &= R_y' - M_y \\ c &= M_x - C_x \\ d &= M_y - C_y \end{aligned} .$$

The point R can now be obtained from (4.16) by substituting the value of t' .

Having calculated the point R, the tangent line to the circle at R can be derived. The line L_t is tangent to the circle and passes through R, which is expressed in implicit form:

$$(x - R_x)(R_x - C_x) + (y - R_y)(R_y - C_y) = 0 \quad (4.19)$$

The main reason for calculating the line L_t is now explained. For given parameter value θ , a bundle of parabolas can be generated by varying the control point on the line L_q . One remaining problem is to find the largest parabola which is bounded within the circle. A parabola, $F(s)^*$, that is in contact with a line L_t represents a close approximation to the one that is precisely in contact with the circle. Exact calculation for the contact with the circle is algebraically complicated in that a quartic equation in s is derived and the coefficients of the equation contain t' . To avoid such complexity an approximation in which the parabola is in contact with the line L_t is introduced.

The parabola, $F(s)$, defined by points S, q and T using Bézier's definition (3.4) is

$$\begin{cases} x = (S_x - 2q_x + T_x)s^2 + 2(q_x - S_x)s + S_x \\ y = (S_y - 2q_y + T_y)s^2 + 2(q_y - S_y)s + S_y \end{cases} \quad (4.20)$$

where $q, (q_x, q_y)$, is yet unknown and, since q is on the line $L_q(t')$, q is substituted by (4.16). Thus (4.20) can be expressed as follows:

$$\begin{cases} x = a_1 t'^2 - (a_1 t' - b_1)s + S_x \\ y = a_2 t'^2 - (a_2 t' - b_2)s + S_y \end{cases} \quad (4.21)$$

$$\text{where } a_1 = -2(R_x' - M_x)$$

$$a_2 = -2(R_y' - M_y)$$

$$b_1 = T_x - S_x$$

$$b_2 = T_y - S_y \quad .$$

The parameter t' indicates the location of q along the line $L_q(t')$.

Eliminating x and y from (4.19) and (4.21) yields

$$At's^2 - (At' - B)s + C = 0 \quad (4.22)$$

$$\text{where } A = a_1(R_x - C_x) + a_2(R_y - C_y)$$

$$B = b_1(R_x - C_x) + b_2(R_y - C_y)$$

$$C = (S_x - R_x)(R_x - C_x) + (S_y - R_y)(R_y - C_y) \quad .$$

Since the parabola, $F(s)^*$, and the line L_t is in contact, double roots in s are the only possible solutions. Thus the discriminant of (4.22) should vanish:

$$A^2 t'^2 - 2A(B+2C)t' + B^2 = 0 \quad (4.23)$$

and the meaningful solution of t' is the one that is greater than the other (which is explained in the next section):

$$t' = \frac{A(B+2C) + 2(A^2(BC+C^2))^{1/2}}{A^2} \quad (4.24)$$

The point Q can now be obtained by substituting the solution t' into (4.16). The line L_q has been defined by parameter t' which is based on points M and R' . The line L_q is then redefined by points M and Q and the associated parameter is indicated by t as in Fig. 4-3. Hence $R'(R'_x, R'_y)$ and t' are replaced by $Q(Q_x, Q_y)$ and t in (4.16).

One can note that the valid range of parameter t' of $L_q(t')$ varies as the parameter θ varies. By reformulating the line L_q with M and Q in (4.16), where R' and t' are replaced by Q and t , the valid range of t can be set to $0 \leq t \leq 1$. Thus the valid range of parameters t and θ are $0 \leq t \leq 1$ and $0^\circ \leq \theta \leq 360^\circ$. By varying parameters θ from 0° to 360° and t from 0 to 1, all parabolas within the workspace are generated. If the range of θ and t in which the corresponding parabolas intersect with obstacles is plotted in parameter space (θ, t) , an image of the obstacle is produced. This process is the geometric mapping.

For given value of θ and t , the point Q can be obtained following the process from (4.14) to (4.24) and the corresponding parabola can then be obtained easily by using (4.16) and (4.21) in which R' , t' should be substituted by Q and t . Unfavorable features of parabolic curves such as sharp curvature when parameter value θ is near 0° or 180° should be avoided during geometric mapping. The shortest path or the safest path can be extracted from the clear area of the parameter space. The shortest paths have the smallest value of t from the clear area of parameter space (θ, t) , while the safest paths have the widest clearance from neighboring edges of obstacle images.

t_1 and t_2 , and the determination of parameter interval is explained in Section 4.2.3.

Contact between an obstacle vertex and a parabola

The vertex P_3 is contact with parabola F_2 , Fig. 4-4. Expressions for the parabola, which is essentially the same as (4.21), is repeated here with minor changes in coefficients a_1 and a_2 where R_x' and R_y' are replaced by Q_x and Q_y , respectively,

$$\begin{cases} x = a_1 t s^2 - (a_1 t - b_1) s + S_x \\ y = a_2 t s^2 - (a_2 t - b_2) s + S_y \end{cases} \quad (4.25)$$

$$\text{where } a_1 = -2(Q_x - M_x)$$

$$a_2 = -2(Q_y - M_y)$$

$$b_1 = T_x - S_x$$

$$b_2 = T_y - S_y \quad .$$

Since the parabola is defined in terms of parameter t , t_2 for F_2 can be calculated by substituting P_3 into (4.25). Elimination of t from both equations yields

$$s = \frac{(a_1 \cdot (P_{3y} - S_y) - a_2 \cdot (P_{3x} - S_x))}{(a_1 \cdot b_2 - a_2 \cdot b_1)} \quad (4.26)$$

Parameter s represents a point along the parabola. Parameter t is obtained by substituting (4.26) into (4.24) or (4.25) provided that the denominator of either (4.24) or (4.25) does not vanish. If the solutions of s and t are $0 \leq s \leq 1$ and $0 \leq t \leq 1$, contact between vertex P_3 and parabola F_2 occurs.

Contact between an obstacle edge and a parabola

The line segment defined by P_1 and P_4 contacts parabola F_1 . Calculation of parameter t_1 is similar to the one given previously in

Section 4.2.1 where the contact between a parabola and line L_t yields a solution for t in (4.24). The edge defined by vertices P_1 and P_4 , in parametric form, is

$$\begin{cases} x = (P_{1x} - P_{4x}) \cdot u + P_{4x} \\ y = (P_{1y} - P_{4y}) \cdot u + P_{4y} \end{cases} \quad (4.27)$$

Elimination of x and y from equations (4.25) and (4.27) yields

$$\begin{cases} (P_{1x} - P_{4x}) \cdot u + P_{4x} = a_1 \cdot t \cdot s^2 - (a_1 \cdot t - b_1) \cdot s + S_x \\ (P_{1y} - P_{4y}) \cdot u + P_{4y} = a_2 \cdot t \cdot s^2 - (a_2 \cdot t - b_2) \cdot s + S_y \end{cases} \quad (4.28)$$

and again elimination of u from (4.28) yields

$$Ats^2 + (-At + B)s + C = 0 \quad (4.29)$$

where $A = (P_{1y} - P_{4y})a_1 - (P_{1x} - P_{4x})a_2$

$$B = (P_{1y} - P_{4y})b_1 - (P_{1x} - P_{4x})b_2$$

$$C = (P_{1y} - P_{4y})(S_x - P_{4x}) - (P_{1x} - P_{4x})(S_y - P_{4y})$$

Equation (4.29) has double roots in s , since the parabola is just in contact with the edge of the obstacle. Thus the discriminant of (4.29) will vanish:

$$A^2t^2 - 2(AB + 2AC)t + B^2 = 0 \quad (4.30)$$

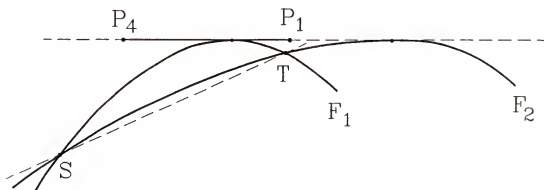


Fig. 4-5 Two possible cases of contact between a line and a parabola

Equation (4.30) may have two different solutions for t as shown in Fig. 4-5 in which F_1 and F_2 are the corresponding curves. It reveals that the root for F_2 approaches zero and is always smaller than the other root for F_1 . The root for F_1 is the valid solution due to the fact that the line segment considered here, P_1P_4 , is bounded by the triangle SQT, Fig. 4-4. Hence t is given by

$$t = \frac{A(B+2C) + 2(A^2(BC+C^2))^{1/2}}{A^2} \quad (4.31)$$

Having calculated t , the corresponding s can be obtained by substituting (4.31) to (4.29),

$$s = \frac{AC + (A^2(BC+C^2))^{1/2}}{A(B+2C) + 2(A^2(BC+C^2))^{1/2}} \quad (4.32)$$

The parameter u can now be calculated from one of the equations of (4.28) provided that the denominator does not vanish. If the values of s , t and u are all between zero and one, a contact occurs. If edge P_1P_4 is parallel to the line $L_q(t)$, the denominator of (4.31) vanishes and t becomes infinite. This special case is considered in the following section.

Special cases

When Q lies on line ST , the parabola collapses to a straight line which is collinear with line ST . This case is discussed in Section 4.1 in which intersections of the straight path ST with edges of obstacles are considered.

When considering a contact of a vertex or an edge with parabolas, singular cases associated with the values of s and t , such as $t=0$, $t=\infty$ and t or s being indeterminate may arise. If an edge of an obstacle under consideration lies on the line $G(u)$, B and C vanish from equation (4.29), thus t becomes zero from (4.31) and s becomes indeterminate from (4.32).

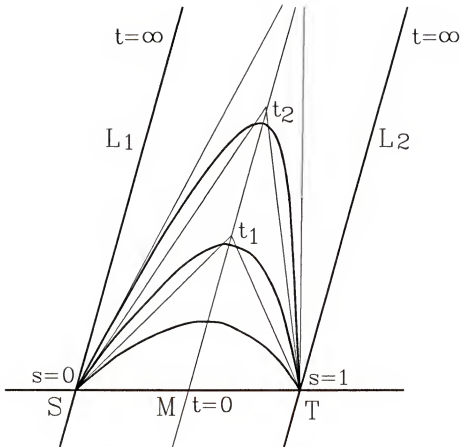


Fig. 4-6 Special cases

The intersection can be checked by considering the vertices of the edge. If one or both of the vertices are inside of line segment ST , intersection occurs.

If an edge under consideration lies on the line L_1 or L_2 in which they are parallel to L_q and pass through S or T , the coefficient A vanishes from (4.29) and t becomes infinite from (4.31). This situation can be avoided by evaluating the value A first, and determining the case. A simple and general method to avoid this situation is to compare the denominator with the numerator at the beginning. If the numerator is less

than the denominator, value of t from (4.31) is in the range $-1 < t < 1$ and it should be evaluated, otherwise the evaluation can be skipped. If t is in the range of $0 < t < 1$, an actual intersection occurs.

If a vertex under consideration lies on the line L_1 or L_2 , s becomes 0 or 1, respectively, from the equation (4.26) and t becomes infinite from equation (4.24) and (4.25). If a vertex point is coincident with S or T , t becomes indeterminate. Such cases can be avoided by comparing the vertex point with S and T .

4.2.3 Ranges of t by each Obstacle

Obstacles in the workspace limit the range of parameter t . Since the calculation of the intersection between an obstacle and the triangle SQT is much simpler than that of between an obstacle and parabolas, obstacles which intersect or are contained entirely within the triangle SQT can be sorted out to selectively apply previously derived formulae in Section 4.2.2. Thus the computation to get the ranges of t can be reduced significantly.

Fig. 4-7 shows all possible cases of interferences between triangle SQT and an obstacle. The calculation of the intersection between an edge of an obstacle and an edge of triangle SQT is carried out by using equations (4.3), (4.4) and (4.5).

In case (a) of Fig. 4-7, the obstacle does not intersect the line segment ST , SQ and QT , hence it may be inside or outside of the triangle SQT . It can be determined by the signs of (4.5) in which points representing S , T and P_0 should be substituted appropriately. For example, a vertex P_0 is determined inside if $|STP_0|$, $|QSP_0|$ and $|TQP_0|$ are positive. If signs for all vertices are positive, it is inside of SQT , otherwise it

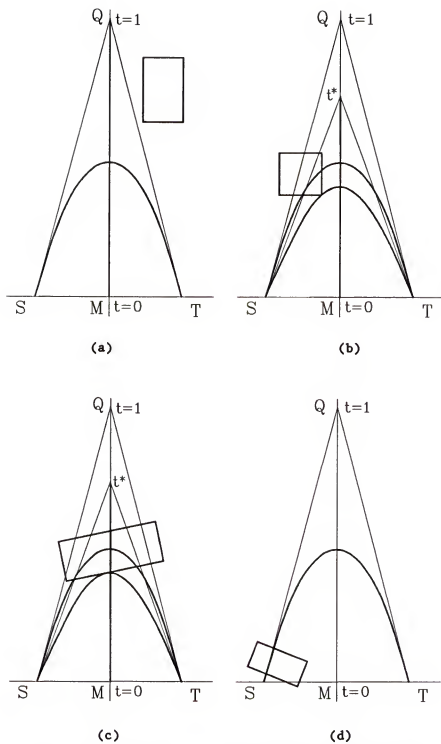
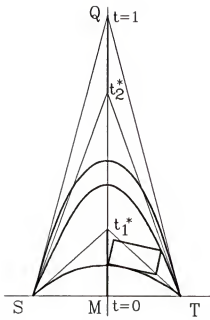
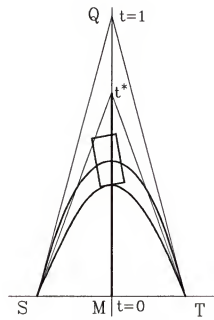


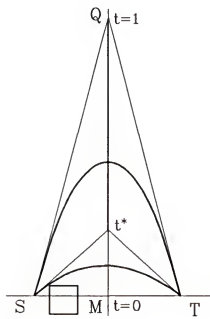
Fig. 4-7 Possible cases of obstacle interference



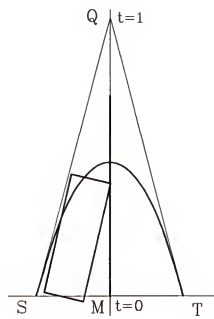
(e)



(f)



(g)



(h)

Fig. 4-7--continued.

is outside of SQT. This case is outside of SQT, and does not require calculation of the range of t because the obstacle is outside of SQT.

In case (b) of Fig. 4-7, the obstacle intersects line segment SQ, hence the interference range of t of the parabolas is $t^* \leq t \leq 1$. It requires calculation of both point and line contact between parabola and obstacle, which is given in the previous section. An obstacle intersecting line segment QT can be treated in the same way.

In case (c) of Fig. 4-7, the obstacle intersects both line segment SQ and ST, therefore, the interference range of t of the parabolas is $t^* \leq t \leq 1$. Calculation is the same as case (b).

In case (d) of Fig. 4-7, the obstacle intersects both line segment ST and SQ, therefore, the entire parabolas intersect with the obstacle. An obstacle intersecting both line segments ST and TQ can be treated in the same way.

In case (e) of Fig. 4-7, the obstacle does not intersect the line segments ST, SQ or QT, rather it is contained in the triangle SQT. The equation (4.5) can be used to determine whether the obstacle is contained in the triangle SQT. A proper substitution of vertex to equation (4.5) is necessary to determine the case. The interference range of t of the parabola is $t_1^* \leq t \leq t_2^*$. It requires calculation of both point and line contact between parabola and obstacle.

In case (f) of Fig. 4-7, the obstacle does not intersect any line segment ST, SQ or QT. It is similar to case (e) except that the resulting value of t_2^* is greater than one, so the interference range of t is $t^* \leq t \leq 1$.

In case (g) of Fig. 4-7, the obstacle intersects line segment ST only. The interference range of t of the parabolas is $0 \leq t \leq t^*$. It requires

computation of point contact between the vertices that are inside the triangle SQT and parabolas.

In case (h) of Fig. 4-7, the obstacle intersects line segment ST only. It is similar to case (g) except that the parameter t^* is greater than one, so all the parabolas intersect with the obstacle.

If more than one obstacle is inside the workspace and the family of parabolas intersects with more than one obstacle, union of the range of t representing intersection by each obstacle is the actual intersection of parabolas with all obstacles inside the workspace. Fig. 4-8 shows the intersection of parabolas with two obstacles, where θ^* indicates the family of parabola. The ranges of t in which parabolas interfere with obstacle A and B are $0 \leq t \leq t_1$ and $t_2 \leq t \leq t_3$. Thus the valid range of t is $t_1 \leq t \leq t_2$ and $t_3 \leq t \leq 1$. Because the number of obstacles that intersect with parabolas varies as the family of parabolas, which depends on parameter θ , varies, the routine for finding the valid range of t should cope with any number of obstacles.

The process described so far creates ranges of t , in which parabolas intersect with obstacles A and B, for given values of θ^* . In Fig. 4-9, the dashed line corresponds to θ^* and values of t_1 , t_2 and t_3 are plotted accordingly. To create whole images of obstacles in parameter space, ranges of t for different values of θ have to be calculated. Such a process creates images in parameter space as shown in Fig. 4.9.

4.2.4 Selection of a Collision-Free Parabola

The process of creating images of obstacles in parameter space is called geometric mapping. One should note that geometric mapping for the entire range of θ is not necessary because the values of θ near 0° and

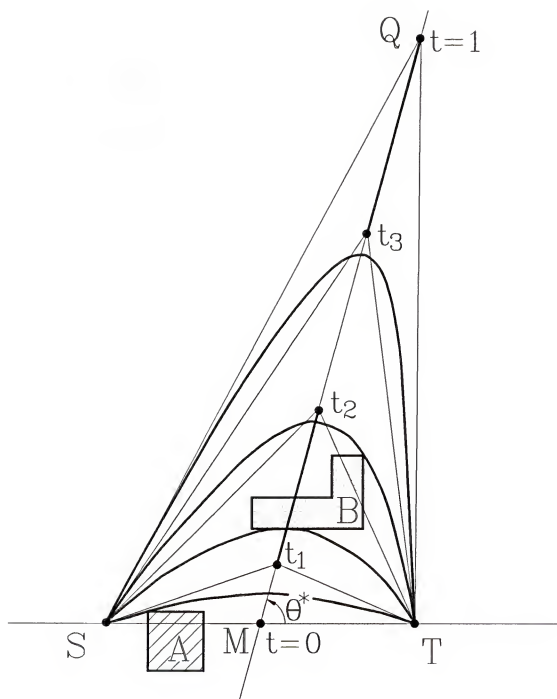


Fig. 4-8 Interference range of t due to two obstacles

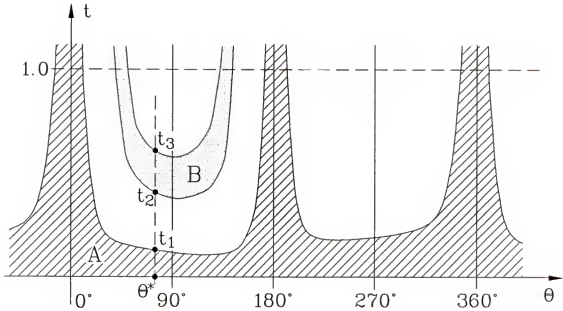


Fig. 4-9 Images of obstacle in parameter space

180° produce sharp parabolas and the valid range of t is much smaller than for other values of θ .

Any point from the clear area of parameter space corresponds to a collision-free parabolic path. It is apparent that selecting a point that is farthest away from boundaries of all obstacle images gives the safest path. The ranges of t for each different value of θ are compared to determine the widest range of t , and the mid value of that range is selected as the value of t . This pair is only an approximation of the safest path, since the comparison is made in θ -direction only. Search for the safest path may not be necessary because any point in the clear area of the parameter space is already a collision-free parabolic path. If obstacles are not exactly where they should be, the choice of path should be the safest one.

It is also possible to select the shortest path in terms of parabolic path by searching for a point that has the smallest value of t . In essence geometric mapping converts the problem of selecting a path in Euclidean space to the problem of selecting a point in parameter space, which is obviously an easier problem. The method of selection can be changed depending on the application.

As an example, the process of selecting the shortest path is now explained. Initially, a vertical line is drawn in parameter space at some value, for example θ^* , and the smallest value of t is found, t_1 , from a vertical line-segment that belongs to the clear area. By increasing the value of θ , another value of t can be obtained and compared to the previous value t_1 . The pair (θ, t) having smaller value of t is stored. This process is continued until θ^*+360° , and the pair (θ, t) having the smallest value of t is obtained. By substituting θ to (4.15), $R'(R'_x, R'_y)$ is obtained and substitution of R' and t to (4.21) yields the shortest parabola.

4.3 Composite Path

Fig. 4-10 shows a situation in which obstacles in the workspace do not permit a parabola to pass from S to T without intersections. In such a case parameter space will be filled with obstacle images. It is then required to use a higher degree Bézier curve other than a parabolic curve, however, as the degree of curve increases so does the number of control points that define the curve. Since each control point associates two parameters the total dimension of parameter space increases to two times the number of control points. Theoretically, the geometric mapping in

dimensions higher than two is possible, but it is computationally too complicated to implement.

An alternative to employing a higher order curve for geometric mapping is to connect parabolic curves in series and smooth the connection points. Thus the original problem is reduced to one of finding intermediate connection points, and the smoothing of the slope discontinuities at connection points. Once intermediate points are found, the geometric mapping is applied to get a collision-free parabola for each subinterval. Slope discontinuity at connection points between two parabolas can be made smooth after connecting collision-free parabolas. These issues are discussed in the following subsections.

4.3.1 Selection of Intermediate Connection Points

A connection point cannot be selected arbitrarily. It should be within the workspace and outside of obstacles. It is obvious that a connection point should be selected in such a way that the resulting path should circumvent at least those obstacles that block the straight path from start to target point. Intermediate points can be selected at surrounding areas of those obstacles and geometric mapping is carried out for each subinterval to obtain a collision-free path. Fig. 4-10 shows a method to select an intermediate connection point. The vertices P_2 and P_3 are on one side and P_1 and P_4 are on the other side of line ST. P_2 , P_3 and P_1 , P_4 are viable points for paths on the upper and lower sides, respectively. P_2 and P_1 are closer to the perpendicular bisector of ST, so connection points selected around P_2 and P_1 tend to produce paths of similar length. Although an intermediate connection point is selected near vertex P_2 and the resulting path is close to the vertex, the smoothing

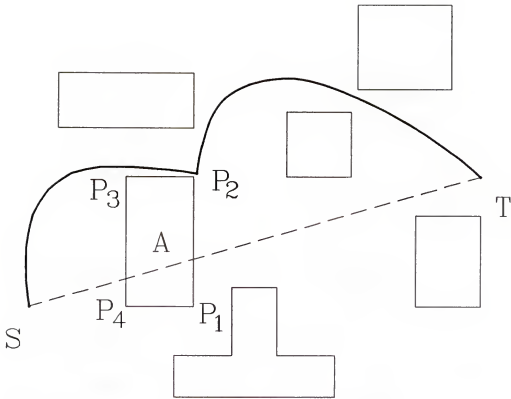
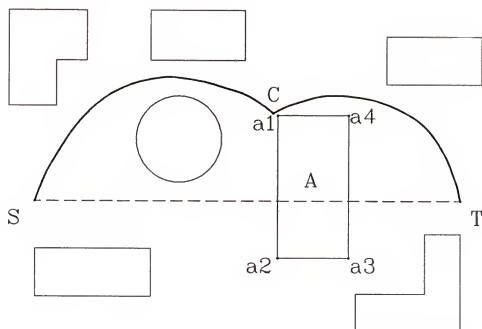


Fig. 4-10 Selection of intermediate connection point

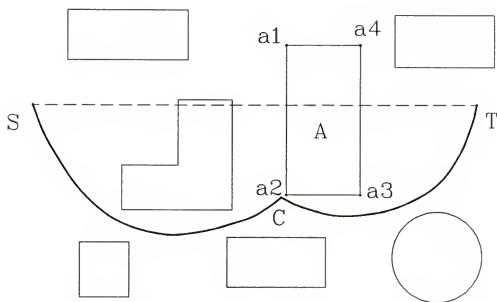
process yields a path that has clearance between path and obstacle as shown in Fig. 4.14. This point will be explained in section 4.3.3.

4.3.2 Four Fundamental Modes of Paths

In general, it is desirable to classify most fundamental ways of connection regardless of the number of intermediate connection points. Thus implementation of the algorithm can be simplified. Four fundamental paths are shown in Fig. 4-11. Another reason for the categorization is to give room for the smoothing process so that the possibility of interference between the altered portion of the path and obstacles is reduced as much as possible.



(a)



(b)

Fig. 4-11 Four fundamental mode of paths

(a) Mode Ia; (b) Mode Ib; (c) Mode IIa; (d) Mode IIb;

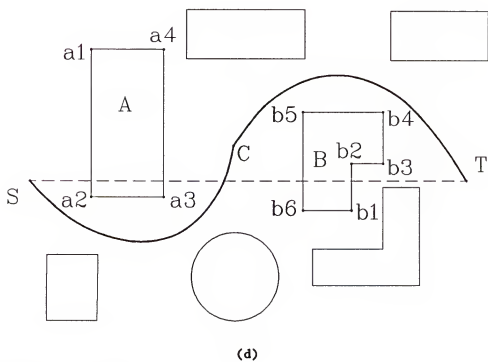
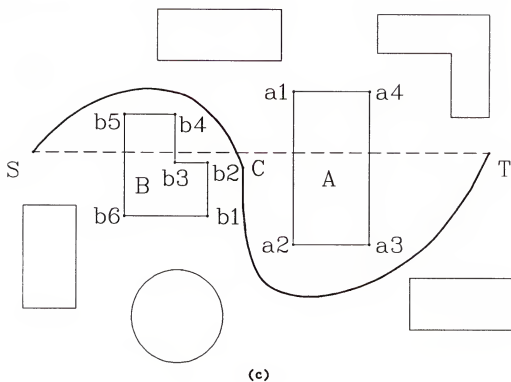


Fig. 4-11--continued.

The four fundamental modes of paths are mode Ia, mode Ib, mode IIa and mode IIb. Subintervals of fundamental modes of paths consist of collision-free paths. The collision-free paths for subintervals are the results from geometric mapping. In the process of geometric mapping, the parameter range of θ is divided into two ranges $0^\circ \leq \theta \leq 180^\circ$ and $180^\circ < \theta \leq 360^\circ$. A parabola with low value of θ indicates the range of θ being $0^\circ \leq \theta \leq 180^\circ$ and a parabola with high value of θ indicates the range of θ being $180^\circ < \theta \leq 360^\circ$. In mode Ia, two parabolas with low values of θ are connected in a series. In mode Ib, two parabolas with high values of θ are connected in a series. In mode IIa, a parabola with low value of θ and a parabola with high value of θ are connected in a series. In mode IIb, a parabola with high value of θ and a parabola with low value of θ are connected in a series.

It is also critical to check if the selected intermediate point is inside of any other obstacle. If it is, the selected intermediate connection point should be discarded. If a collision-free path for a subinterval by a fundamental mode is found to be impossible to create, other modes are employed. It is also possible that all path modes fail to generate a collision-free path for a certain subinterval. Such a case indicates that geometric mapping for that subinterval does not provide a collision-free path. Therefore it has to be divided into smaller subintervals.

There are variations of mode I and mode II to provide additional modes for the case when the chosen mode fails to generate collision-free paths. The purpose of additional modes is to prevent unnecessary division of the subinterval. Fig. 4-12 shows three alternative paths for mode Ia

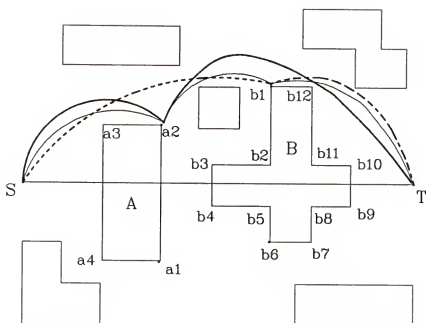


Fig. 4-12 Variations of mode Ia

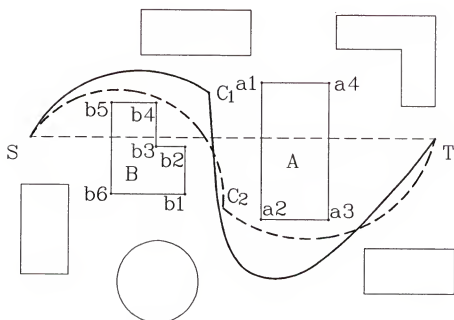


Fig. 4-13 Variations of mode IIa

where more than two obstacles are across the line segment ST and an intermediate connection point is selected from each obstacle. Fig. 4-13 shows two alternative paths for the mode IIA, where two intermediate points are selected at the center of line segments a_1b_4 and a_2b_1 .

4.3.3 Smoothing Slope Discontinuity at Connection Point

If a collision-free path so constructed has a slope discontinuity at a connection point, smoothing is desirable. Four different methods of making a smooth connection of two parametric curves are described in Section 3.1.4. Of the four methods of smoothing slope discontinuity, connection by combination of portions of neighboring curves, equations

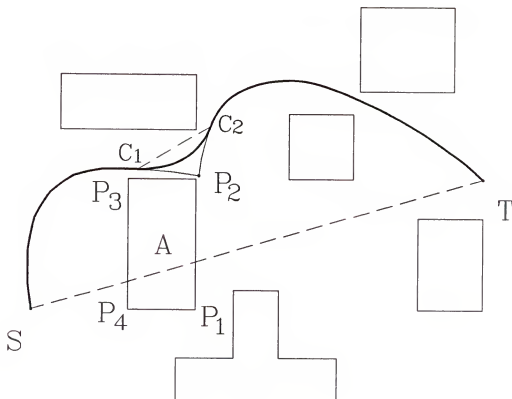


Fig. 4.14 Smoothing slope discontinuity

(3.20) through (3.24), and connection by a quartic curve, equations (3.35) through (3.39), are adopted here. For mode I paths, smoothing by combination of portions of neighboring curves is better because it gives more clearance from obstacles than other methods, as shown in Fig. 4-14. For mode II paths, both methods can be used, because unsmoothed path has a relatively smooth connection.

One should recognize that the smoothing by combination of portions of neighboring curves can destroy the collision-free nature of the original path. The line segment C_1C_2 is used to check for obstacle interference. If it intersects with any obstacle, then the portions C_1P_2 and C_2P_2 are reduced until no interferences occurs between the line segment C_1C_2 and obstacles.

4.4 Collision-Free Paths in Three Dimensions

Key ideas for planning collision-free paths in three dimensions were explained in Section 3.3. It is assumed that obstacles are convex hulls and consist of flat surfaces, for example a cylinder is represented by a polygonal prism. A comprehensive treatment on the topics related to convex hulls can be found in a book by Preparata and Shamos(1985).

The flat projection plane may be defined by three points, start, target and the origin of the global coordinate system. A local coordinate system can be set on the projection plane, Fig. 4-15. The x and y axes of the local coordinates are coplanar with the projection plane. The parameter ϕ indicates the amount of rotation about the x axis. A transformation matrix, A, which transforms a vector between the local to the global coordinate systems, can be expressed in terms of unit vectors

that lie along the x, y and z axes of local coordinates.

$$A = \begin{bmatrix} i_x & j_x & k_x \\ i_y & j_y & k_y \\ i_z & j_z & k_z \end{bmatrix} \quad (4.33)$$

where $i = (i_x \ i_y \ i_z)$, $j = (j_x \ j_y \ j_z)$ and $k = (k_x \ k_y \ k_z)$ are the unit vectors. Since the matrix A itself is a orthogonal matrix, transpose of A, A^T , transforms a vector from global coordinates to the local coordinate system.

There are two ways of calculating intersection of obstacles with the projection plane. One is to carry out the transformation of all vertices to the local coordinate system, then calculate the intersections with the x-y plane of the local coordinate system. The other is to determine intersections of obstacles with the projection plane expressed in global coordinates. The resulting intersection points are transformed to local coordinates to exclude the z component. Computation of intersections has to be done for all edges of the obstacles.

As an illustration, the process of obtaining point P_1 of Fig. 4-16 is given as follows. The edge connecting v_0 and v_1 intersects the projection plane and the intersection point is P_1 . Transformation of v_0 and v_1 in global coordinates to local coordinates can be done by multiplying the transformation matrix A^T to obtain

$$\begin{aligned} v_0' &= A^T v_0 \\ v_1' &= A^T v_1 \end{aligned} \quad (4.34)$$

The edge connecting v_0' and v_1' , in parametric form, is

$$\begin{aligned} x(t) &= (1-t)v_0'_x + tv_1'_x \\ y(t) &= (1-t)v_0'_y + tv_1'_y \end{aligned} \quad (4.35)$$

$$z(t) = (1-t)v_{0z}' + tv_{1z}',$$

and the projection plane expressed in local coordinates is

$$z = 0 \quad (4.36)$$

since the x and y axes are attached to the projection plane. Thus the edge and projection plane are both expressed in the local coordinate system. By substituting (4.36) into (4.35) the parameter t can be obtained:

$$t = \frac{v_{0z}'}{v_{0z}' - v_{1z}'} \quad (4.37)$$

Substitution of (4.37) into (4.35) yields the coordinates of intersection point P_1 .

Obstacle images on the projection plane, however, are not yet complete. The computation merely provides intersection points between edges of obstacles and the projection plane. These points have to be organized to form a convex polygon, because intersection of a convex hull with a flat plane produces a convex polygon. A method of forming a convex polygon from those points on the projection plane is now explained for Fig. 4-16 as an example. An indepth treatment on this subject can be found in Preparata and Shamos (1985). Vertex numbering in Fig. 4-16 indicates the sequence of calculation of intersections which depends on the numbering of vertices of Fig. 4-15. Point P_1 and P_2 are the reference points and form a line. Remaining points are divided into two groups depending on if they are on the left or right side of the reference line. Two points from one side are selected again as reference and the division separates remaining points into two groups again. This process stops when the number of points on one side is less than two. For instance, in Fig.

4-16 point P_1 and P_2 are reference points and points P_3 and P_4 are on the left side. Point P_4 is on the right side with respect to the line drawn between points P_1 and P_3 . The points are stored in a tree structure. Points retrieved from left to right of the tree structure are organized in a clockwise fashion; those from right to left are ordered in an anticlockwise manner. The final result can be either in the sequence P_1, P_3, P_4, P_2 or P_1, P_2, P_4, P_3 . The image can now be fed into the geometric mapping.

To get a collision-free path in three dimensional environments, control points for a two dimensional collision-free path, P in Fig. 4-16, have to be converted back to three dimensions. This can be done by multiplying the two dimensional control points by transformation matrix A . This reverses the process transforming a vertex from global to local coordinates. Translations between the origins of the three dimensional workspace and the two dimensional workspace should be properly set to match the three dimensional collision-free path.

4.5 Implementation of the Algorithms

Shown in Fig. 4-17 is an overall flow of logic for finding a two dimensional collision-free path. Search for a collision free-path stops when either a successful path is found or the recursion depth becomes zero which indicates a failure. For three dimensional environments, a section for finding two dimensional images of obstacle on projection plane should be placed just after the input data.

Algorithms are implemented on a VAX 11/750 computer and a Silicon Graphics IRIS 4D/70 graphics workstation. The program is developed in C

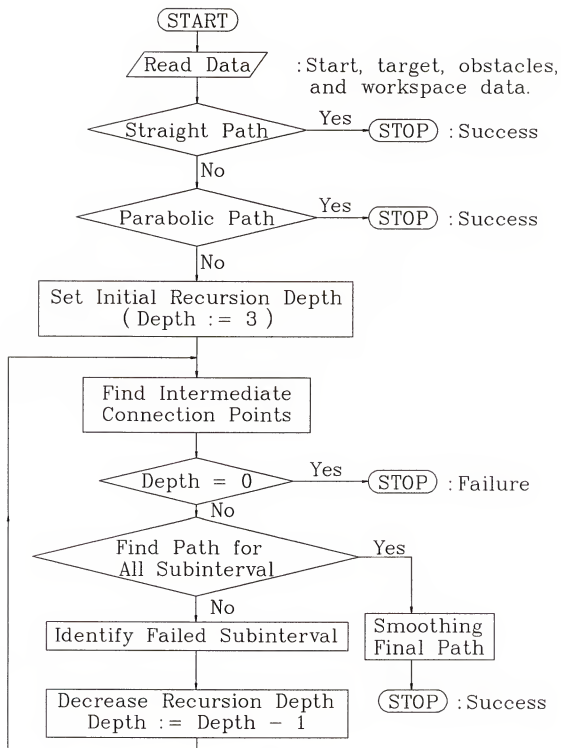


Fig. 4-17 Flow chart of two dimensional collision-free path algorithm

language due to the portability between the VAX 11/750 and the Silicon Graphics IRIS 4D/70. Input parameters for the program are origin, start and target points, radius of the work space, coordinates of obstacle vertices and the expansion length. Obstacle information is stored in structure variables, a feature in C language, and each obstacle is organized in a linked list so that the number of obstacles and the number of vertices in an obstacle are not limited. Adding an obstacle, therefore, is accomplished by adding the information to the data file.

Simulation results are given in Appendix A for three different environments: randomly scattered obstacles, obstacles resembling compartments and obstacles resembling rooms and corridors. Simulations were carried on VAX 11/750 and plotted using Hewlett-Packard 7475A plotter. Confining walls are represented by obstacles too, hence the rectangular area inside of the walls represents the actual work area. Various shaped polygons represent obstacles and dash-dot lines indicate expanded obstacles. When a collision-free parabola between start and target points is not found, intermediate connection points are searched. A collision-free path for a sub-interval is plotted with a dash-dot line. The final path after the smoothing is plotted with a solid line.

Execution time of the program depends on several factors: total number of obstacles, location and shape of obstacles, algorithm implementation methodology, and computer to be used. Approximate figure of execution time for the plots given in Appendix A is: 0.1(sec) for Fig. A-1, 0.3 (sec) for Figures A-2, A-3 and A-7, 0.5 (sec) for Figures A-4, A-5 and A-6, 1.0 (sec) for Figures A-8 through A-11.

4.6 Concluding Remarks

In this chapter the detailed algorithms for planning collision-free paths have been studied. A straight path connecting the start and target point is examined at the beginning to check if any obstacle is across the path. If it is, a two dimensional geometric mapping based on parametric parabola is conducted and images of obstacles in parameter space is constructed. To have a proper relationship between Euclidean space and parameter space, the ranges of parameters have to be set initially. Control points of a parabola are organized in polar coordinates and ranges of θ and t are set to $0^\circ \leq \theta \leq 360^\circ$ and $0 \leq t \leq 1$. For varying values of θ , intersections of parabola with obstacles are calculated in terms of t . This produces images of the obstacle in parameter space and a point from the clear area of parameter space corresponds to a collision-free parabola. If parameter space is filled with images of obstacles, a collision-free parabola connecting start and target point does not exist. At this point, instead of using higher order curves, intermediate connection points are determined. Then, the geometric mapping for the subsections divided by an intermediate point is carried out. Subsequently, smoothing the slope discontinuity at the connection point is conducted.

For collision-free paths in three dimensional environments, a straight path connecting start and target point is examined at the beginning to check if any obstacle is across the path. If the straight path intersects an obstacle, a three dimensional geometric mapping is used to obtain a collision-free path. Three parameters, ϕ , θ and t , are involved in three dimensional parametric parabolas. To get a three

dimensional geometric mapping, two dimensional geometric mapping is applied to the direction of ϕ . In other words, by varying the value of ϕ and applying two dimensional geometric mapping in the θ - t plane a three dimensional geometric mapping results.

Geometric mapping based on a curve of degree higher than two is not fully investigated in this dissertation. Parameter space produced by such a geometric mapping may ultimately be filled with images of obstacles when the workspace is cluttered, therefore, requiring intermediate connection points. Higher order curves, however, reduce the necessity of finding intermediate connection points.

CHAPTER 5
COLLISION AVOIDANCE OF INDUSTRIAL ROBOTS
IN CONJUNCTION WITH PLANNING COLLISION-FREE PATHS

In this chapter problems associated with the collision avoidance of a planar 3R robot and a General Electric P60 robot are discussed. Collision avoidance here refers to a class of problems in which a path for the robot is predefined and the robot maneuvers to escape collision. Another class of problems are that the start and target points are given and the robot maneuvers to avoid collision. Work related to the first can be found in Young and Duffy (1987), in which a planar 3R robot was examined. In this the robot avoids obstacles by obtaining the allowable range of orientation for every step along the path and changing the orientation to the mid-value of the allowable range. A second can be found in Brooks (1983b), in which a PUMA robot was examined. The first three joints of the PUMA robot were assumed active. A collision-free path is obtained in joint space¹ which ensures collision-free motion of the robot in Euclidean space.

The first class of problem is considered in this chapter. A path is assumed to be given by previously developed path planning algorithm. Several algorithms which include collision recognition, obstacle expansion and change of configuration are developed for collision avoidance. The

¹ The dimension of a joint space is the same as the number of joints the robot has. A 3R robot has a three-dimensional joint space.

assumptions made are that orientation of the end-effector can be altered along the path, however, position cannot be changed during the change of configuration.

One of the main objectives of this chapter is to develop a method for identifying collision of a robot. It is desirable to identify a collision before the robot moves to the next point. Obstacles that may be machines, workpieces, machine operators, other robots or any objects inside the robot workspace reduce the actual workspace of the robot. Related work to the analysis of robot workspace may be found in Kumar, A. and Waldron (1981), Sugimoto and Duffy (1981) and Davidson and Hunt (1987). The motion of a robot should be restricted in a designated space to avoid accidents. Collision recognition has not been incorporated in robot control loops, rather it is left to robot operators. Hence operators have to consider preventing collisions in controlling robots. Collision recognition may be added to existing robot control loops as an option to help reduce the programming time as well as prevent accidents. A related work on collision recognition can be found in Crane (1987) in which one obstacle (a sphere or a cube) was assumed. Collision detection was carried out in real time utilizing clipping hardware of a graphics computer (Silicon Graphics IRIS series computer) in which the obstacle was replaced by a bounding box. Any attempt to cross the box sets the flag which indicates that clipping occurs. Collision recognition discussed in section 5.2. is machine independent and is able to cope with any number of obstacles. Collision recognition is carried out by testing interferences of robot link with obstacles in which robot links are represented by line segments.

Collision-free paths are obtained through the geometric mapping discussed in Chapters 3 and 4 with the assumption that a point moves along the paths. Since a robot end-effector cannot be assumed to be a point, modifications to obstacles have to be made so that the final path can maintain enough clearance from obstacles. One approach to this problem is to expand obstacles by an amount which is greater than one-half of the width of the end-effector. In a related work, expansion of obstacles is described in Lozano-Pérez and Wesley (1979) as grown obstacles. In their work obstacles are grown depending on the reference point and orientation of the object to be moved. In section 5.1, expanding two and three dimensional obstacles in equal amount in all directions is considered.

A collision-free path created by considering the expansion of obstacles still has limitations in guiding robotic manipulators. Since the base joint of a robot manipulator is attached to the ground, any of the other links may hit an obstacle or other links while the end-effector moves along the path safely. To identify such a situation, collisions have to be tested before the robot moves to the next step. Other limiting cases which occur are when the path is out of the reachable workspace or out of the robot actuator range. An additional algorithm for change of configuration is introduced to overcome these situations. When a link is about to hit an obstacle or a joint angle is too close to the actuator limitation, the configuration of the robot is changed by rotating the end-effector orientation to alleviate the problem while maintaining the end-effector position. In some situation, this scheme of change of configuration may not solve the problem without altering the end-effector position. A study for change of configuration considering both altering

the orientation and the position of the end-effector is not attempted, and left for future research.

A complete solution for the problems associated with collision avoidance can be achieved by obtaining paths in joint space, Brooks (1983b). He demonstrated such collision-free motion for a PUMA robot where the first three joints are active. In his work, obstacles in Euclidean space are mapped into three dimensional joint space and a collision-free path is searched through connectivity graphs.

Similar to this, a geometric mapping could be carried out in joint space; then the resulting path guarantees a collision-free motion for the robot. However, obtaining the joint space is computationally too expensive to use in real-time control of a robot. Control of a planar 3R robot and a General Electric P60 robot are considered in sections 5.4 and 5.5, respectively.

5.1 Expansion of Obstacles

5.1.1 Expansion of Two Dimensional Obstacles

Fig. 5-1 shows the expansion of a two dimensional obstacle. The original obstacle is indicated by solid lines, and an expanded and shrunken obstacle is indicated by dashed lines. It is assumed that the numbering of vertices is counterclockwise in the following derivation. A proper translation of edges is important. For instance, L_1' is the translation of L_1 to the right and L_1'' is translated to the left. The intersection of L_1' and L_4' produces a vertex P_1' of the expanded obstacle and the intersection of L_1'' and L_4'' produces a vertex P_1'' of the shrunken obstacle. The line L_1 represents the edge defined by vertices P_1 and P_2

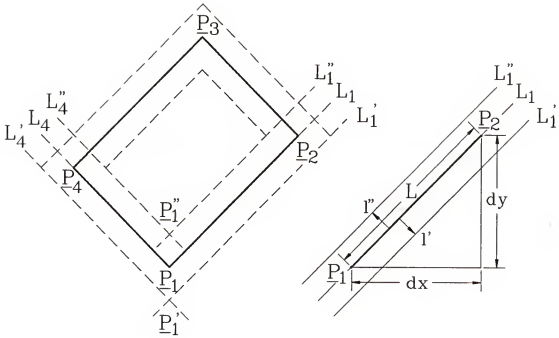


Fig. 5-1 Expansion of an obstacle

and is expressed in parametric form:

$$L_1 : \begin{cases} x = (P_{2x} - P_{1x})t + P_{1x} \\ y = (P_{2y} - P_{1y})t + P_{1y} \end{cases} \quad (5.1)$$

Translational terms are necessary to express lines L_1' and L_1'' by adding these terms to equation (5.1). For L_1' , the translational term $l'(l_x', l_y')$ is given by

$$\begin{aligned} l_x' &= \frac{1'}{L} (P_{2y} - P_{1y}) \\ l_y' &= -\frac{1'}{L} (P_{2x} - P_{1x}) \end{aligned} \quad (5.2)$$

where l' is the perpendicular distance between L_1 and L_1' and L is the distance between P_1 and P_2 . For L_1'' , the translational term $l''(l_x'', l_y'')$ is given by

$$\begin{aligned}
 l_x'' &= -\frac{1}{L} (P_{2y} - P_{1y}) \\
 l_y'' &= \frac{1}{L} (P_{2x} - P_{1x})
 \end{aligned} \quad (5.3)$$

Addition of translational terms into equation (5.1) yields

$$L_1' : \begin{cases} x = (P_{2x} - P_{1x})t + P_{1x} + l_x' \\ y = (P_{2y} - P_{1y})t + P_{1y} + l_y' \end{cases} \quad (5.4)$$

$$L_1'' : \begin{cases} x = (P_{2x} - P_{1x})t + P_{1x} + l_x'' \\ y = (P_{2y} - P_{1y})t + P_{1y} + l_y'' \end{cases} \quad (5.5)$$

If vertices are arranged clockwise, the signs of translational terms should be made opposite to obtain properly expanded or shrunken obstacles. No assumptions are made on the shapes of obstacles hence the derivation works for either convex or concave obstacles.

5.1.2 Expansion of Three Dimensional Obstacles

Expansion of three dimensional obstacles is similar to that of two dimensional cases. Fig. 5-2 shows expansion of a three dimensional obstacle where the expansion amount is given by length l . Vector $\underline{a}$ is perpendicular to the surface, S_1 , which is defined by vertices P_1 , P_2 , P_3 and P_4 and is expressed by

$$\begin{aligned}
 \underline{a} &= (P_2 - P_1) \times (P_3 - P_1) \\
 &= (a_x, a_y, a_z)
 \end{aligned} \quad (5.6)$$

The magnitude of vector $\underline{a}$ is denoted by $|\underline{a}|$. Surface S_1' , defined by vertices P_1' , P_2' and P_3' can be expressed in implicit form:

$$S_1' = A_1x + B_1y + C_1z - D_1 = 0 \quad (5.7)$$

where

$$A_1 = \begin{vmatrix} 1 & 1 & 1 \\ P_{1y}' & P_{2y}' & P_{3y}' \\ P_{1z}' & P_{2z}' & P_{3z}' \end{vmatrix} \quad B_1 = \begin{vmatrix} 1 & 1 & 1 \\ P_{1z}' & P_{2z}' & P_{3z}' \\ P_{1x}' & P_{2x}' & P_{3x}' \end{vmatrix}$$

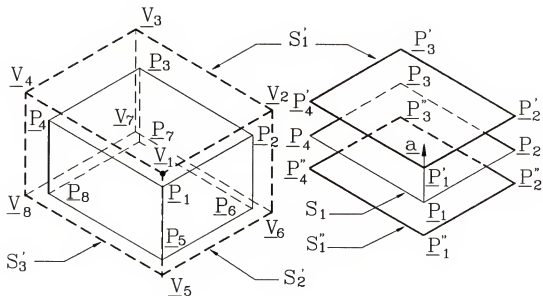


Fig. 5-2 Expansion of a three-dimensional obstacle

$$C_1 = \begin{vmatrix} 1 & 1 & 1 \\ P_{1x}' & P_{2x}' & P_{3x}' \\ P_{1y}' & P_{2y}' & P_{3y}' \end{vmatrix} \quad D_1 = \begin{vmatrix} P_{1x}' & P_{2x}' & P_{3x}' \\ P_{1y}' & P_{2y}' & P_{3y}' \\ P_{1z}' & P_{2z}' & P_{3z}' \end{vmatrix}.$$

The vertices for S_1' and S_1'' are obtained by translating the vertices of S_1 . The translational terms for S_1' are given by

$$\begin{aligned} l_x &= \frac{a_x}{|a|} l \\ l_y &= \frac{a_y}{|a|} l \\ l_z &= \frac{a_z}{|a|} l \end{aligned} \quad (5.8)$$

The vertices of the surface S_1' , P_1' , P_2' , P_3' and P_4' , are obtained by adding translational terms to vertices P_1 , P_2 , P_3 and P_4 . Likewise the

vertices of the surface S_1'' , P_1'' , P_2'' , P_3'' and P_4'' , are obtained by subtracting translational terms from P_1 , P_2 , P_3 and P_4 respectively.

A vertex of expanded obstacle V_1 is a common intersection of three surface S_1' , S_2' and S_3' . The surfaces S_2' is defined by P_1' , P_2' and P_5' , and the surface S_3' is defined by P_1' , P_4' and P_5' .

$$S_2' = A_2x + B_2y + C_2z - D_2 = 0 \quad (5.9)$$

$$S_3' = A_3x + B_3y + C_3z - D_3 = 0 \quad (5.10)$$

Coordinates of $V_1(V_{1x}, V_{1y}, V_{1z})$ can be obtained by solving simultaneous equations (5.7), (5.9) and (5.10). A proper substitution of vertices in the above formulae is essential because the numbering of the vertices of a surface is random, unlike two dimensional cases.

5.2 Collision Recognitions

5.2.1 Two Dimensional Collision Recognition

Shown in Fig. 5-3 is a collision between a link and an obstacle. Dashed line indicate the expansion of the original obstacle where the expansion amount is one-half of the width of the robot links, so links can then be represented by straight line-segments. Links and edges of obstacles are expressed in parametric equations. Intersections between these equations determine collisions. Derivations for determining collision between link L_2 and an edge V_1V_2 is given as an illustration. The points P_1 and P_2 are the end points of the link L_2 . The link L_2 is represented by equation $G(u)$ and the edge V_1V_2 is given by $H(v)$:

$$G(u) \begin{cases} x(u) = (P_{2x} - P_{1x})u + P_{1x} \\ y(u) = (P_{2y} - P_{1y})u + P_{1y} \end{cases} \quad (5.11)$$

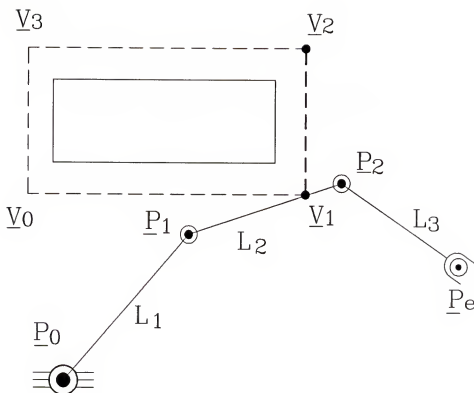


Fig. 5-3 Collision between a link and an expanded obstacle

$$H(v) \begin{cases} x(v) = (V_{2x} - V_{1x})v + V_{1x} \\ y(v) = (V_{2y} - V_{1y})v + V_{1y} \end{cases} \quad (5.12)$$

At intersection, $G(u) = H(v)$ and the solutions for u and v are as follows:

$$u = \frac{(V_{1y} - P_{1y})(P_{2x} - P_{1x}) - (V_{1x} - P_{1x})(P_{2y} - P_{1y})}{(V_{2x} - V_{1x})(P_{2y} - P_{1y}) - (V_{2y} - V_{1y})(P_{2x} - P_{1x})} \quad (5.13)$$

$$v = \frac{(V_{2y} - V_{1y})(P_{1x} - V_{1x}) - (V_{2x} - V_{1x})(P_{1y} - V_{1y})}{(V_{2x} - V_{1x})(P_{2y} - P_{1y}) - (V_{2y} - V_{1y})(P_{2x} - P_{1x})} \quad (5.14)$$

When the solution for v of equation (5.14) is in the range $0 \leq v \leq 1$, the edge of the obstacle does intersect the line passing through P_1 and P_2 , however it does not necessarily intersect with the line segment P_1P_2 . In addition, if the solution for u of equation (5.13) is in the range

$0 \leq u \leq 1$, the edge of the obstacle intersects the line segment representing link L_2 . If the denominator of (5.14) vanishes, edge V_1V_2 and link L_2 are parallel or collinear; such special cases can be considered as non-intersecting. The calculation has to be carried out for all edges of the obstacles to determine the intersections between the obstacles and robot links.

5.2.2 Three Dimensional Collision Recognition

Collision between a link of a robot and obstacles can be identified in a way similar to the planar case. The difference is that intersections are between surfaces of obstacles and the line representing links. In Fig. 5-4, the obstacle is expanded by one-half of the width of a robot link. The surface $A(s,t)$ defined by vertices V_0 , V_1 , V_2 and V_3 can be expressed in a bilinear parametric equation:

$$A(s,t) = [(1-s) \quad s] \begin{bmatrix} V_0 & V_3 \\ V_1 & V_2 \end{bmatrix} \begin{bmatrix} (1-t) \\ t \end{bmatrix} \quad (5.15)$$

where the parameter range of s and t are $0 \leq s \leq 1$ and $0 \leq t \leq 1$. The line $L(u)$ representing a robot link is given by

$$L(u) = (P_3 - P_2) u + P_2 \quad (5.16)$$

where the valid interval of parameter u is $0 \leq u \leq 1$ for the line segment P_2P_3 . The condition for finding the intersection point of the surface $A(s,t)$ and the line $L(u)$ is

$$A(s,t) = L(u) \quad (5.17)$$

After substituting equations (5.15) and (5.16) into (5.17) and rearranging the equation yields:

$$A_x s t + B_x s + C_x t + D_x u + E_x = 0$$

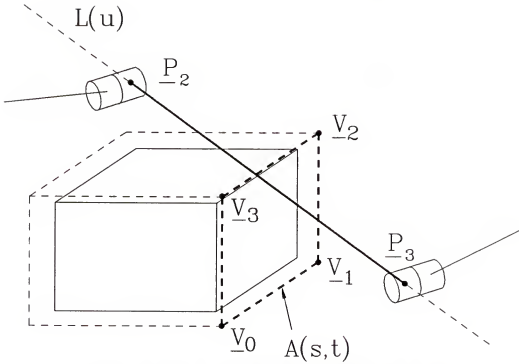


Fig. 5-4 Collision between a link and an expanded obstacle

$$A_y s t + B_y s + C_y t + D_y u + E_y = 0 \quad (5.18)$$

$$A_z s t + B_z s + C_z t + D_z u + E_z = 0$$

$$\text{where } A_x = V_{0x} - V_{1x} + V_{2x} - V_{3x}$$

$$B_x = -V_{0x} + V_{1x}$$

$$C_x = -V_{0x} + V_{3x}$$

$$D_x = P_{2x} - P_{3x}$$

$$E_x = V_{0x} - P_{2x}$$

A_y , A_z , B_y , B_z , C_y , C_z , D_y , D_z , E_y and E_z are calculated similarly by substituting the y and z components of P_2 , P_3 , V_0 , V_1 , V_2 and V_3 .

Eliminating u from equation (5.18) results in equations in s and t :

$$a_x s t + b_x s + c_x t + d_x = 0$$

$$a_y s t + b_y s + c_y t + d_y = 0$$

(5.19)

$$\begin{aligned}
\text{where } a_x &= A_x D_z - A_z D_x & a_y &= A_y D_z - A_z D_y \\
b_x &= B_x D_z - B_z D_x & b_y &= B_y D_z - B_z D_y \\
c_x &= C_x D_z - C_z D_x & c_y &= C_y D_z - C_z D_y \\
d_x &= E_x D_z - E_z D_x & d_y &= E_y D_z - E_z D_y .
\end{aligned}$$

Elimination of t leads to a quadratic equation in s :

$$(a_y b_x - a_x b_y) s^2 + (b_x c_y - c_x b_y + a_y d_x - a_x d_y) s + (c_y d_x - c_x d_y) = 0 . \quad (5.20)$$

Since there is only one intersection between a line and a plane surface, the discriminant of (5.20) will vanish. The double roots of s are given by:

$$s = \frac{b_x c_y - c_x b_y + a_y d_x - a_x d_y}{2 (a_x b_y - a_y b_x)} \quad (5.21)$$

and a corresponding solution for t can be obtained from one of the equations of (5.19):

$$t = - \frac{b_y s + d_y}{a_y s + c_y} . \quad (5.22)$$

The solution for u can be obtained by substituting the solutions of s and t into one of the equations of (5.18) provided that the denominator does not vanish. If the solutions for s , t and u are in the ranges $0 \leq s \leq 1$, $0 \leq t \leq 1$ and $0 \leq u \leq 1$ an intersection occurs, since $0 \leq s \leq 1$ and $0 \leq t \leq 1$ represents the surface patch defined by vertices V_0 , V_1 , V_2 and V_3 of the obstacle and $0 \leq u \leq 1$ represents line segment ST .

If the denominator of equation (5.21) or (5.22) vanishes, the surface and the line are parallel or coplanar and such a case can be considered as non-intersecting. The calculation has to be carried out for all sides of the obstacles and links to determine collision. One should see that the derivations in this and the previous sections are

significantly simpler using the parametric form of equations than using explicit or implicit forms of equations.

5.3 Change of Configurations

As described at the introduction of this chapter, change of configuration is used to avoid collision with obstacles and to stay within the range of joint actuators. An assumption is made that the end-effector orientation may change along the path. Whenever a robot is about to hit any obstacles or reach joint actuator limits, the orientation is changed in such a way that the robot changes configuration to alleviate the problem while maintaining the original end-effector position. One should note that the change of configuration assumes that the orientation is a free parameter.

5.3.1 Change of Orientation of Two Dimensional Case

Supposing that a vector $\underline{x}$ indicates orientation of the end-effector and a matrix M indicates rotation of the orientation. Then the orientation vector $\underline{x}'$ after the rotation is given by

$$\underline{x}' = M\underline{x} \quad . \quad (5.23)$$

M may be expressed as follows:

$$M = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \quad (5.24)$$

where θ is the rotation angle measured counterclockwise.

5.3.2 Change of Orientation of Three Dimensional Case

Contrary to the two dimensional case, orientation of the end-effector for the three dimensional case has to be set by two independent

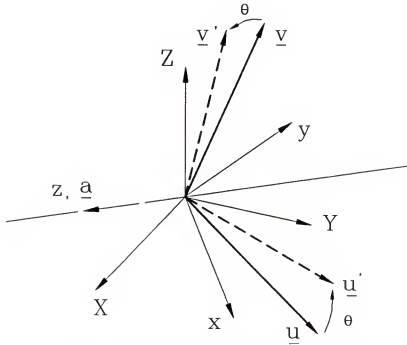


Fig. 5-5 Rotation of orientation vectors

vectors. The vectors $\underline{u}$ and $\underline{v}$ in global coordinates (the X-Y-Z coordinate system) indicate the orientation, Fig. 5-5. Rotation of orientation should be taken on an axis that provides greater dexterity. Such a rotation axis depends on the robot kinematic structure as explained in section 5.5. The axis of rotation may be expressed by a unit vector $\underline{a}$. A local coordinate system (the x-y-z coordinates) is attached to the vector $\underline{a}$ so that the z-axis is aligned with $\underline{a}$. The orientation vectors $\underline{u'}$ and $\underline{v'}$ after rotation about the $\underline{a}$ axis can be expressed as follows:

$$\underline{u'} = \text{TMT}'\underline{u} \quad (5.25)$$

$$\underline{v'} = \text{TMT}'\underline{v} \quad (5.26)$$

where T is an orthogonal transformation matrix and M is the rotation matrix. Matrix T transforms a vector from local coordinates to global coordinates, and the transpose of T, T', transforms a vector from global

coordinates to local coordinates. Matrix T may be obtained from the vector $\underline{a}$:

$$T = \begin{bmatrix} \frac{a_y}{(a_x^2 + a_y^2)^{1/2}} & \frac{a_x a_z}{(a_x^2 + a_y^2)^{1/2}} & a_x \\ -\frac{a_x}{(a_x^2 + a_y^2)^{1/2}} & \frac{a_y a_z}{(a_x^2 + a_y^2)^{1/2}} & a_y \\ 0 & -(a_x^2 + a_y^2)^{1/2} & a_z \end{bmatrix} \quad (5.27)$$

The rotation is taken around the z-axis of the local coordinates, hence the rotation matrix is given by

$$M = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.28)$$

where θ is the amount of rotation in the counterclockwise direction. Details of two and three dimensional transformations can be found in Rogers and Adams (1976).

5.4 Planar 3R Robot

Shown in Fig. 5-6 is a 3R robot work-environment where S and T indicate the start and target points. The large circle indicates the workspace boundary, i.e. the boundary of the extreme reachable workspace¹.

¹ The extreme reachable workspace consists of all points that are accessible to the robot, and the reachable workspace consist of all points that are accessible to the robot for a given orientation, the definition appears in Kumar and Waldron (1981).

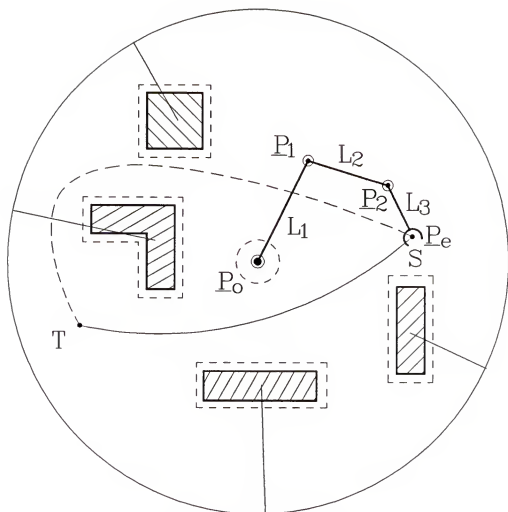


Fig. 5-6 A 3R robot work-environment

The solid lines indicate original obstacles and dashed lines indicate expanded obstacles. The amount of expansion should be greater than one-half the width of the end-effector. The circle in dashed lines surrounding the base joint, P_0 , is an imaginary obstacle that protects the base joint. In some cases it may also represent the inner void of the workspace where the robot cannot reach if the base link is longer than the sum of the remaining links.

Of two paths shown in solid and dashed lines, the path in dashed lines cannot be used because the robot cannot move toward the target point without colliding with the obstacle between the path and the base joint. In order to prevent such a path, imaginary barriers connecting obstacles and the workspace boundary are added which are shown as dashed line segments. In the process of geometry mapping the expanded obstacles, the imaginary barriers and the imaginary circle representing the base joint are all treated as obstacles. Hence the path by a solid line can be obtained from geometry mapping. One may note that the paths resulting from the geometric mapping are always bounded by the limit of extreme reachable workspace.

As explained in the previous section, the configuration of the robot has to be changed in three different occasions: violation of the actuator operating limits, movement beyond the reachable workspace and collision with obstacles. The case in which joint actuator limits are exceeded may be identified by comparing the result of reverse analysis with the valid ranges of the actuators. Movement beyond the workspace may be identified during the reverse analysis,¹ or it may be identified by comparing the results of the forward analysis and the input data to the reverse analysis. Collision with obstacles can be identified by the method given in section 5.2.

The robot is guided along the path with initially given orientation. Once a limit case occurs during the movement, the solutions of the reverse analysis are discarded and the input data for the previous

¹ A thorough treatment on reverse and forward analysis of robotic manipulators can be found in Duffy (1981).

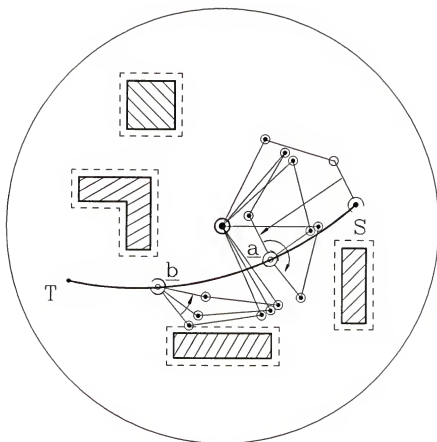


Fig. 5-7 Change of orientation

position are recalled. The motion toward the target point stops and the change of configuration is carried out. The allowable range of orientation should be obtained before changing the orientation. Shown in Fig. 5-7 is two cases of change of orientation. The robot changes orientation at point a because the angle of joint P_1 is out of range, and it changes orientation at point b because links L_2 and L_3 intersect the obstacle.

The allowable range of orientation can be obtained by rotating the orientation in both directions until a limit case occurs, so that the allowable rotation angles are obtained. One should note that the

calculation of the allowable range of orientation is carried out in background without moving the robot. After obtaining the allowable range of orientation, the original orientation is rotated toward the mid-value of the allowable range, and the robot then moves toward the target point. The calculation of allowable range of orientation is required only if the change of configuration is necessary. After the change of configuration, the robot resumes to move toward target point with the new orientation resulting from the change of configuration.

In Young and Duffy (1987), the method of computing allowable range of rotation of a planar 3R robot is via transformation of two line segments representing obstacle edges and the robot link to a regular rectangle. The transformation yields an expression for allowable range of rotation. The transformation is carried out for each link and the common intersection of the allowable range of rotation is the permissible range of rotation. The robot configuration is then determined by the mid-value of permissible range of rotation. This process is repeated each step as the robot moves toward target point.

An animated simulation of collision avoidance of a 3R robot in conjunction with collision-free path generation is developed on a Silicon Graphics IRIS 4D graphics workstation with the UNIX operating system. The C programming language and the graphics library¹ supported by the system are used to build the program. The hardware z-buffering saves considerable time in managing overlapped images. A uniform shading with

¹ Graphics related subjects are not included in this dissertation. One may find excellent treatments of the subject in Rogers (1985) and Foley and Van Dam (1982).

one light source at infinity is employed to produce realistic images. Shown in Fig. 5-8 is a three dimensional modeling of a 3R robot, obstacles and a collision-free path for given start and target points. The program includes interactive control of parameters using a mouse and function keys. The main menu includes change of setting, motions with and without joint limitations and program termination. The submenus of the change of setting include the change of start position, target position, obstacle location and path parameters. The menu items included in the change of path parameter are: shorter path, safer path and increase and decrease parameter t . The scene can be rotated and scaled with the mouse, and translated with the cursor keys. Pressing the function key F1 activates and deactivates the drawing of expanded obstacles in wireframe, and the F2 key is for drawing the path. The escape key terminates the program at any time. Shown in Figures 5-8 through 5-13 are some examples of the program operation.

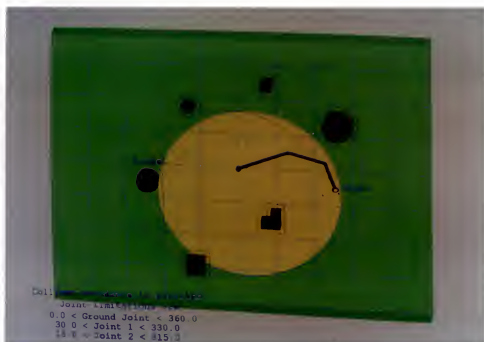


Fig. 5-8 Initial settings and a collision-free path.



Fig. 5-9 A collision-free path after changing start point, target point and location of obstacles

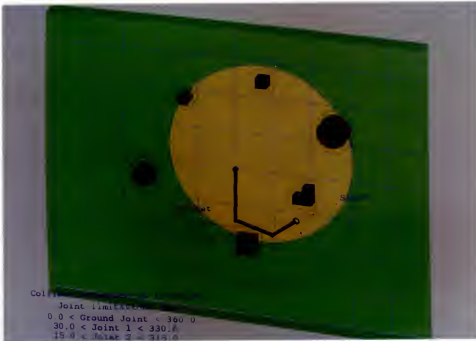


Fig. 5-10 Before the collision with an obstacle
(change of configuration is triggered)

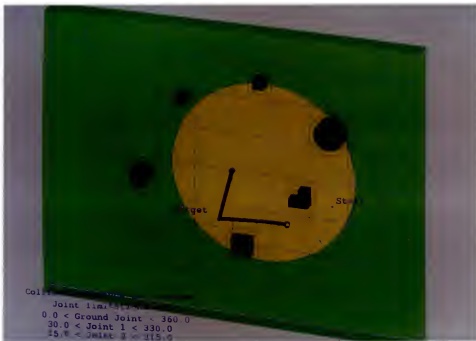


Fig. 5-11 After the change of configuration

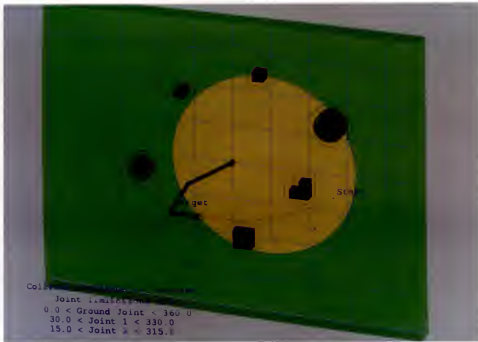


Fig. 5-12 One step before movement beyond the reachable workspace (change of configuration is triggered)

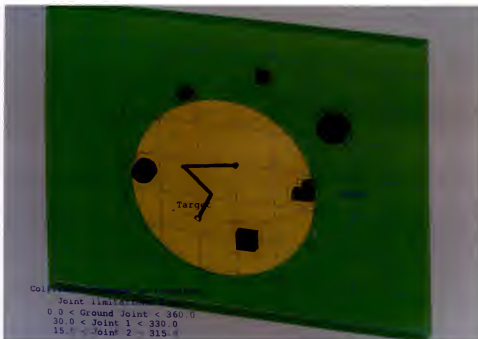


Fig. 5-13 After the change of configuration

5.5 General Electric P60 Robot

The limit cases for a GE (General Electric) P60 robot are basically the same as for the planar 3R robot. Shown in Fig. 5-14 is a kinematic model of a GE P60 robot; the displacement analysis is given in Appendix B. A computationally efficient method for determining if the path is out of the reachable workspace is to use the displacement analysis. Violation of the valid joint-actuator range is determined by observing the results of the reverse analysis. Collision between a link and an obstacle was discussed in section 5.2.2 where the obstacle was expanded and the robot links were represented by line-segments. In order to represent a link as a line-segment, position of end points of each link should be computed. Interference of robot links with obstacles can be monitored by computing intersections between the line segments representing links and surface patches representing obstacles. It is also possible to represent the robot link as polyhedra, such as rectangular prism, and then intersections between edges of links and obstacles are determined. This procedure, however, will take more computation time.

As explained in section 5.3.2, change of configuration is carried out by rotating the orientation vectors without altering the end-effector position. In the case of a planar 3R robot, the axis of rotation is defined by the z-axis where the robot operates in the x-y plane. In the case of the GE P60 robot, the rotation axis should be selected in the direction that provides greater allowable range of rotation; such an axis depends on the configuration of the robot. Instead of finding the axis that provides the greatest allowable range of rotation, several axes which are parallel to S_4 , S_5 and a_{45} and attached to the end-effector are

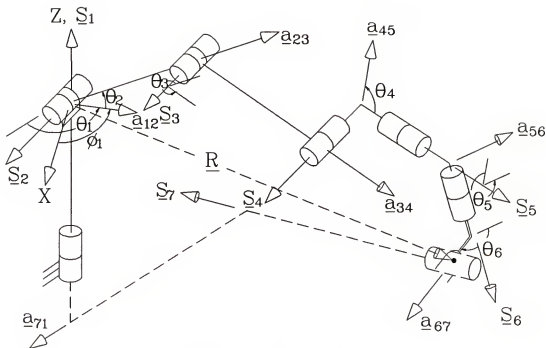


Fig. 5-14 Kinematic model of a GE P60 robot

selected. When a limiting case occurs, that axis which provides the greatest allowable range of orientation is determined from these preselected axes. The allowable range of rotation can be calculated by rotating the orientation in both directions until a limit case occurs and the total rotation angle is the allowable range of orientation. Rotation of orientation may be carried out by substituting a rotation axis vector, for example S_5 , into (5.27) and rotation angle into (5.28), then the resulting orientation vectors can be obtained from (5.25) and (5.26). It is possible that after the change of orientation for the given set of rotation axes, the robot still may not be able to avoid the obstacle, which indicates a failure. Such cases may require altering the end-effector position to avoid obstacles.

The animated simulation of collision avoidance of a GE P60 robot in conjunction with collision-free path generation is carried out on a Silicon Graphics IRIS 4D/70 graphics workstation. Most of the features are similar to that of the planar case. Additional features--toggled by function keys--include display of the projection plane, the path in three dimensions and the windows for projected images, images of obstacle in parameter space and help messages.

Although great amount of time is devoted to develop the simulation program, details of the program are not included here mainly due to the size of program which is over 220 pages. Explanation of the program such as organization of three dimensional object data structures, manipulation of fonts, method of shading, manipulation of windows and transformations of graphic objects would be much greater than the size of the program itself.

Shown in Figures 5-15 through 5-21 are some examples of the operation of the program. Shown in Fig. 5-15 is a GE P60 robot, a collision-free path, a projection plane shaded in transparency, a window for obstacle images on the projection plane and another window for obstacle images in parameter space. Shown in Figures 5-16 and 5-17 are before and after the change of configuration due to imminent collision. Shown in Figures 5-18 and 5-19 are before and after the change of configuration because the second joint is about to exceed the valid range of rotation. A collision-free path after changing the position of the start, target and obstacles is shown in Fig.5-20. Shown in Fig. 5-21 is an example of the change of the start, target and an obstacle.

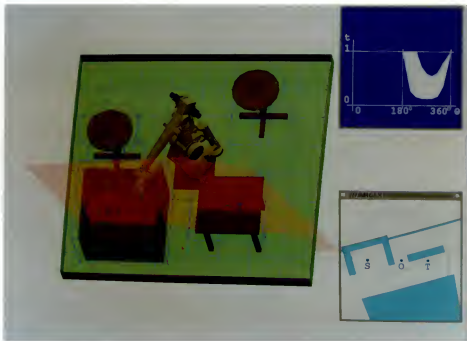


Fig. 5-15 Initial settings with a collision-free path



Fig. 5-16 One step before the collision.
(change of configuration is triggered)



Fig. 5-17 After change of configuration due to collision



Fig. 5-18 One step before the valid range of a joint actuator is exceeded. (change of configuration is triggered)



Fig. 5-19 After the change of configuration



Fig. 5-20 A collision-free path after changing the target position



Fig. 5-21 Two windows are opened to change an obstacle position

5.6 Concluding Remarks

In this chapter, expansion of obstacles, collision recognition and change of configuration for two and three dimensional cases were treated. Obstacles were expanded an equal amount in all directions and no assumptions were made on the shape of obstacles. Collision recognition was carried out by intersections between links and obstacles. In deriving the formulae for intersections, parametric forms of equations were used to simplify the derivation. The collision recognition can be added to existing robot control-loops to reduce the danger of collision and to restrict the motion of robots in a designated space.

A change of configuration was introduced to avoid limit cases such as exceeding the joint-actuator ranges, moving out of the reachable

workspace and collision with obstacles. It was carried out by rotating the orientation vectors while the end-effector stayed at the same position. In the case of a spatial 6R robot, several rotation axes were preselected and one of the axes that provides the greatest allowable range was chosen.

The collision avoidance was developed in conjunction with collision-free path generation. A collision-free path is predefined by geometric mapping and the robot follows the path. If a limit case occurs, the configuration is changed to alleviate the problem without altering the end-effector position. As mentioned in the introduction of this chapter, an absolute collision avoidance can be achieved by planning collision-free path in joint space. Identification of limit cases, therefore, will not be necessary.

An animated simulation of a planar 3R robot and a GE P60 robot were developed on a Silicon Graphics IRIS 4D graphics workstation. The three dimensional solid modeling of the 3R and the GE P60 robot features uniform shading with one light source. The start, target and obstacle position and path parameters can be changed interactively with the mouse and the function keys. The execution time for obtaining a collision-free path depends on the environment settings. Typical examples given in this chapter take less than one-hundred milliseconds. Comparisons among different algorithms require a benchmark settings such as standardized start, target points and obstacle number and shape, and this is left for future work.

The simulation reveals that the execution time depends on the graphics routine rather than displacement analysis and collision

recognition of the robot. Hence a delay at the graphics subsystem occurs as more objects are included in the drawing list and graphics features such as opening several windows, z-buffering, shading and backfacing polygon removal are added in the graphics routine.

CHAPTER 6 CONCLUSIONS

6.1 Planning Collision-Free Paths

Moving a point from one point to another without intersecting any object in a workspace is the fundamental problem of path planning research. The approach taken for planning collision-free paths is via geometric mapping based on Bézier curves of order two. Images of obstacles in parameter space are made by geometric mapping, and a point selected from the clear area corresponds to a collision-free path. When the parameter space is filled with obstacle images, intermediate connection points are searched and a collision-free path for each subinterval is obtained by geometric mapping. This results in a connection of several collision-free paths. Smoothing the connection region is carried out to ensure the slope continuity throughout the path.

The algorithm breaks down if either the start or target point is inside a maze, or if it is enclosed partially (or totally) by overlapping obstacles or concave obstacles. The execution time for geometric mapping depends on how the algorithms are implemented, the computer used and the environment settings such as the number and shape of the obstacles.

6.2 Collision Avoidance of Robot Manipulators

Guiding a robot along a given path while avoiding collisions has been performed. The path is obtained by the previously developed method for collision-free paths. Expansion of obstacles and addition of imaginary obstacles are introduced to exclude the paths that are unacceptable to guide the robot from the process of geometric mapping.

Algorithms for collision recognition are developed to identify intersections between robot links and obstacles and between robot links themselves. The algorithm can be implemented in real-time collision recognition to complement the existing robot control-loops and to reduce the robot programming time.

Change of configuration is developed to alleviate limit situations: collisions, moving out of the reachable workspace and exceeding the valid range of joint-actuators. Configuration is changed by rotating the end-effector orientation while staying at the same position.

Animated simulations of a planar 3R robot and a GE P60 robot are carried out on a Silicon Graphics IRIS 4D graphics workstation. The simulations show that the overhead due to collision recognition, when compared with the same simulation without collision recognition, is negligible for the examples given in Chapter 5.

6.3 Further Considerations

As mentioned in the previous chapters, geometric mapping is based on Bézier curves of order two. An immediate extension of the geometric mapping would be the mapping based on cubic Bézier curves. It may require

more computation, however, a search for intermediate connection points is less likely to be required.

One of the assumptions for geometric mapping is that obstacles are stationary. In the case of moving obstacles, a new geometric mapping may be required to update the path when an obstacle moves and intersects the established path. The revision of the path may depend on the type of robot. For a mobile robot, it may stay in a safe area until the path becomes clear, or a new path from the current point to the target point may be constructed immediately. This area requires much further investigation.

Collision avoidance of robot manipulators is demonstrated with an assumption that the path to follow is given by geometric mapping. Geometric mapping here transforms obstacles in Euclidean space into parameter space. If path planning is carried out in joint space, i.e. geometric mapping transforms obstacle images from joint space to parameter space, maneuvering of the manipulator to avoid limit situations will not be necessary. Developing computationally efficient methods for obtaining obstacle images in joint space and coping with the resulting high dimensionality of geometric mapping are the main problems.

Another area that has not been considered is the comparison of different algorithms for planning collision-free paths. This may require substantial efforts due to the complexity of each algorithm, but will provide an invaluable insight for the future research direction.

Implementation of collision recognition and path planning to the existing robot control loops is yet another area where industrial robots may benefit. Data for obstacles has to be properly entered and displayed

to provide correct feedback to the robot operators. This will require improved I/O handling capabilities of current robot systems.

APPENDIX A
SIMULATIONS OF PLANNING COLLISION-FREE PATHS

Shown in the Figures A-1 through A-11 are some of the results of simulations executed on a VAX 11/750 mini-computer. The output was produced on a Hewlett-Packard 7475A plotter. Expanded obstacles are plotted in dash-dot lines. Long slender rectangles enclosing other small polygons are considered to be walls of the workspace. Three different environment setting are given. Shown in Figures A-1 and A-2 are random settings, Figures A-3 through A-7 are compartment settings and Figures A-8 through A-11 are rooms and corridor settings. The simulations were carried out for several different start and target points.

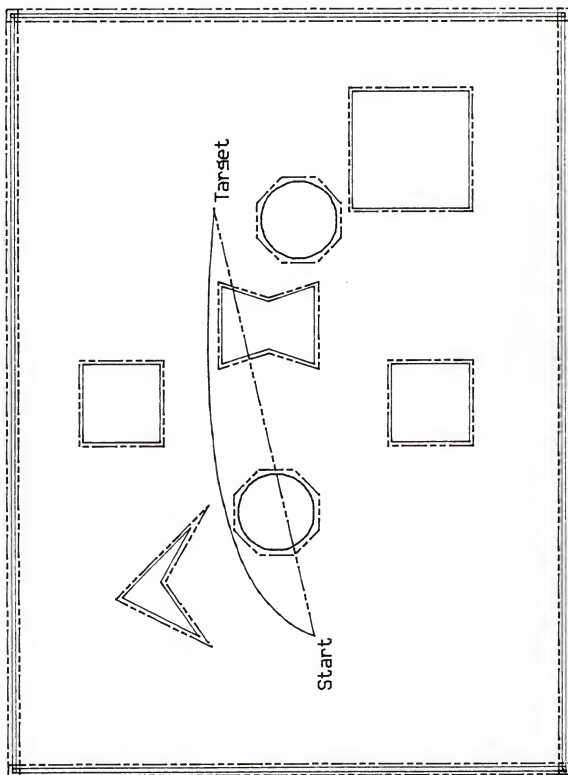


Fig. A-1 Random setting 1

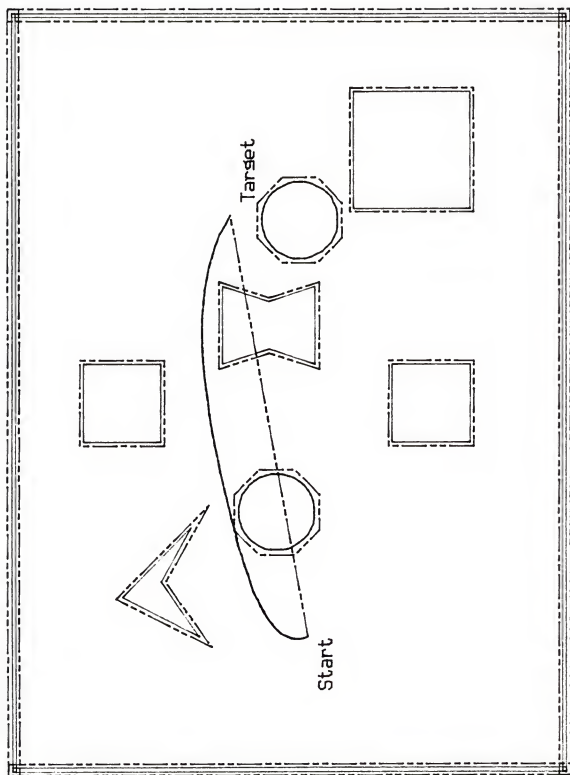


Fig. A-2 Random setting 2

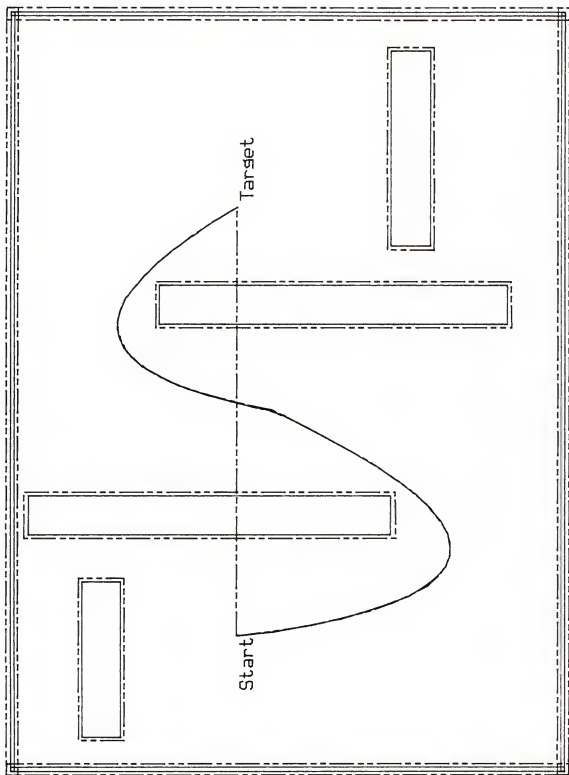


Fig. A-3 Compartment setting 1

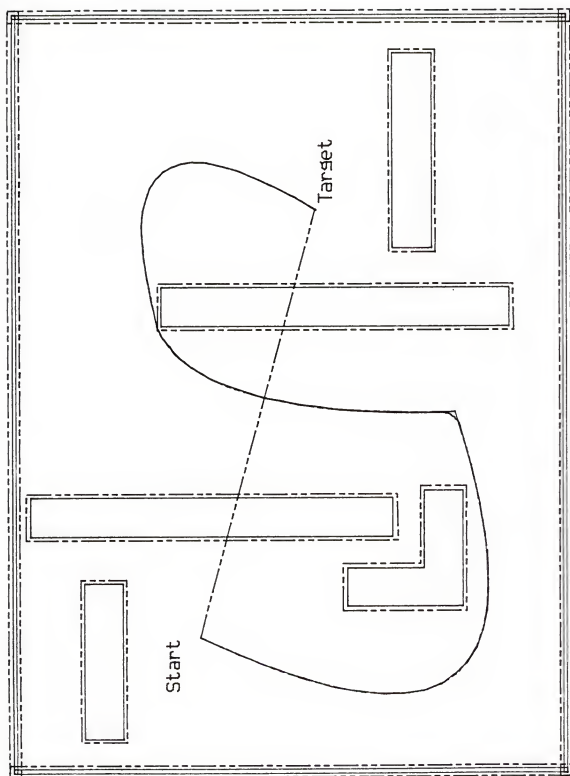


Fig. A-4 Compartment setting 2

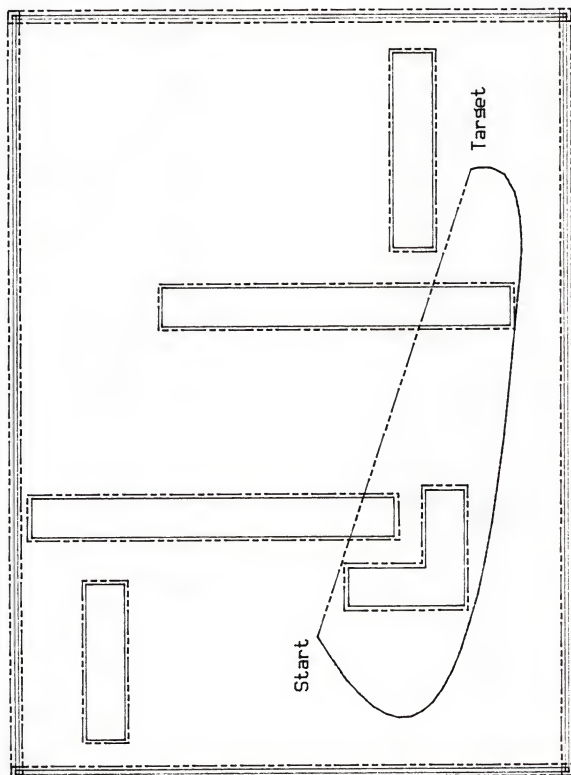


Fig. A-5 Compartment setting 3

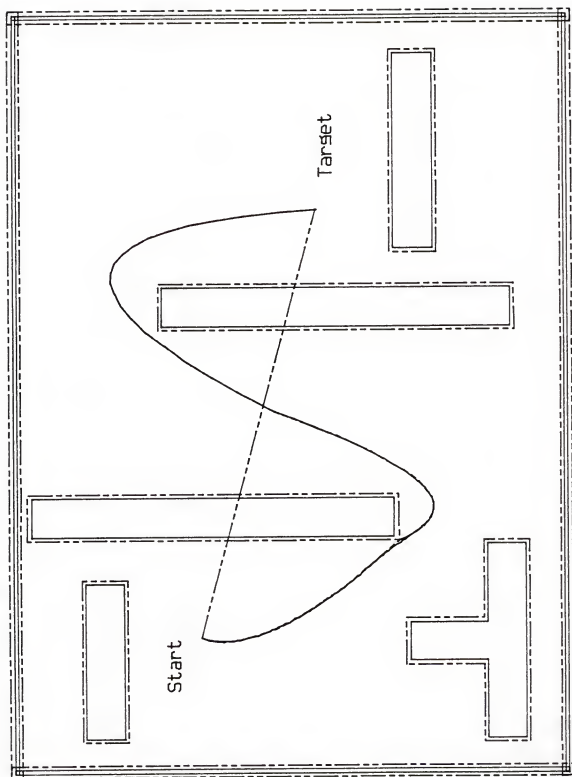


Fig. A-6 Compartment setting 4

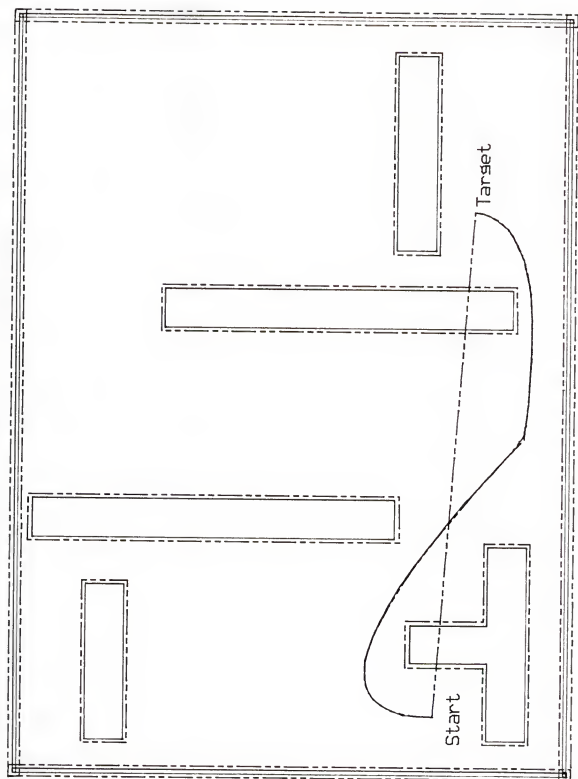


Fig. A-7 Compartment setting 5

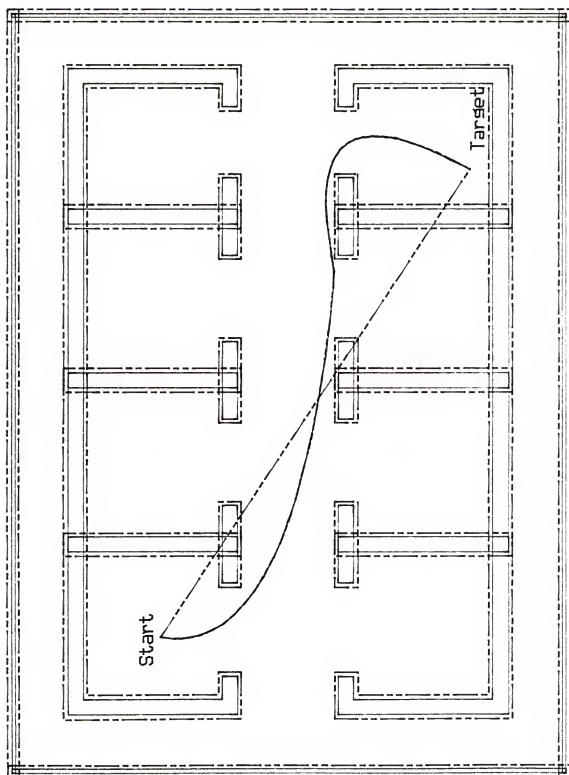


Fig. A-7 Rooms and corridor setting 1

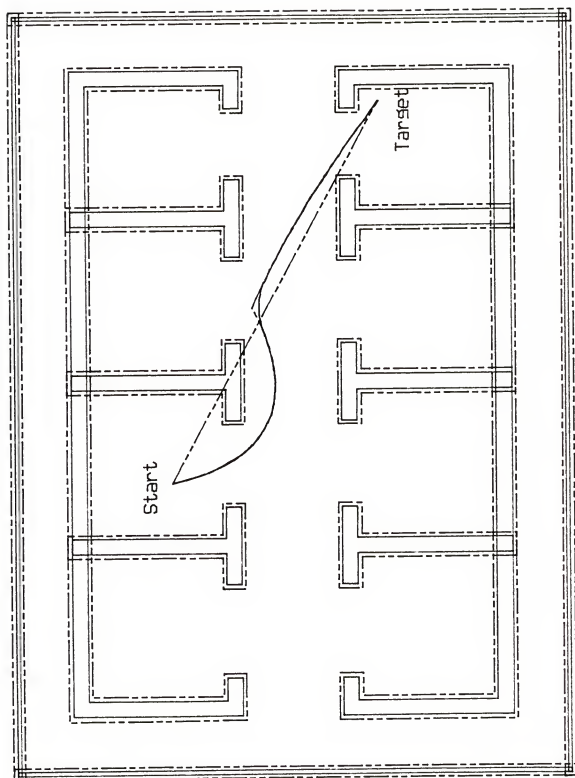


Fig. A-9 Rooms and corridor setting 2

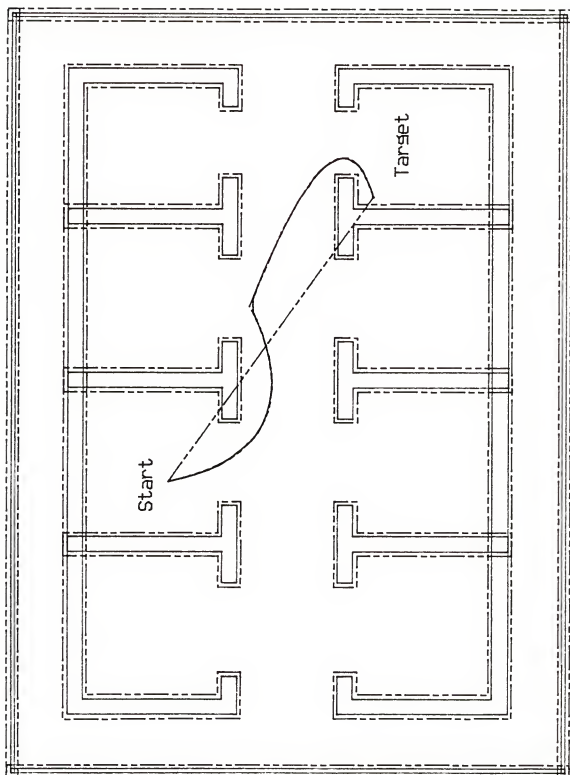


Fig. A-10 Rooms and corridor setting 3

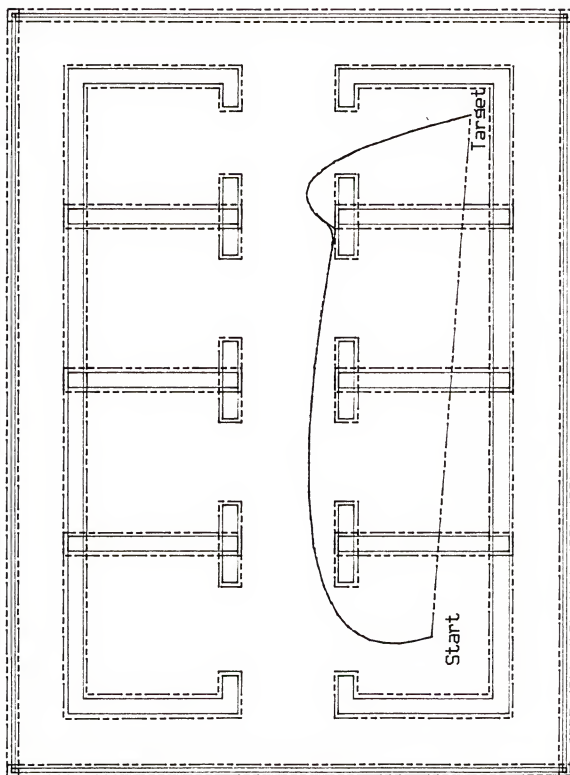


Fig. A-11 Rooms and corridor setting 4

APPENDIX B
DISPLACEMENT ANALYSIS OF THE GE P60 ROBOT

The displacement analysis of the GE P60 robot, which consists of a close-the-loop analysis and the reverse and forward displacement analysis, follows the method established by Duffy (1981). Close-the-loop analysis is for determining the unknown parameters of the hypothetical links connecting the end-effector and the ground joint. The analysis is due to Lipkin and Duffy (1985), and is not repeated here. A GE P60 robot has parallel structures (a four bar mechanism) in links two and three. Analysis of these four bar mechanisms is not necessary if driving the robot is the only concern, however it is necessary for displaying the robot on a graphics computer. The analysis of the four bar mechanisms is not included in this appendix.

Shown in Fig. B-1 is a kinematic model of the GE P60 robot. The hypothetical links ($\underline{S}_7$, $\underline{a}_{71}$) are drawn in dotted lines. Shown in Table B-1 are the parameters associated with the robot with the tool dependent parameters treated as variables. Shown in Table B-2 is the valid range of the joint actuators in which the angles are referenced according to the kinematic model.

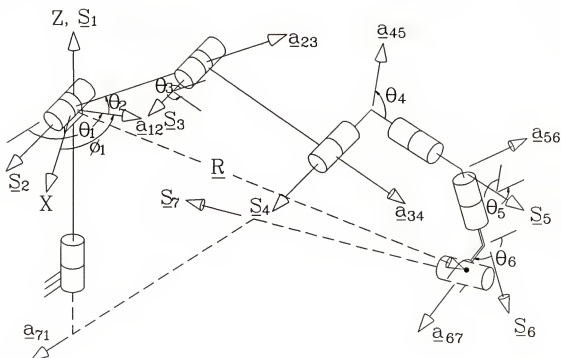


Fig. B-1 Kinematic diagram of a GE P60 robot

Table B-1 Parameters of GE P60 robot

(unit = inch)

$S_{11} = *$	$a_{12} = 0$	$\alpha_{12} = 90^\circ$
$S_{22} = 0$	$a_{23} = 70$	$\alpha_{23} = 0^\circ$
$S_{33} = 0$	$a_{34} = 90$	$\alpha_{34} = 0^\circ$
$S_{44} = 9.8$	$a_{45} = 0$	$\alpha_{45} = 90^\circ$
$S_{55} = 14.5$	$a_{56} = 0$	$\alpha_{56} = 90^\circ$
$S_{66} = 15.4$	$a_{67} = **$	$\alpha_{67} = **$
$S_{77} = *$	$a_{71} = *$	$\alpha_{71} = *$

* : determined from close-the-loop analysis

** : tool dependent parameters.

Table B-2 The valid range of joint actuators

Joint 1 (ground joint)	$-150^\circ \leq \phi_1 \leq 150^\circ$
Joint 2 (shoulder joint)	$44^\circ \leq \theta_2 \leq 130^\circ$
Joint 3 (elbow joint)	$-132^\circ \leq \theta_3 \leq -70^\circ$
Joint 4	$-20^\circ \leq \theta_4 \leq 200^\circ$
Joint 5	$-180^\circ \leq \theta_5 \leq 180^\circ$
Joint 6	$-300^\circ \leq \theta_6 \leq 300^\circ$

B.1 Reverse Displacement Analysis of GE P60 Robot

The reverse displacement analysis is a process for obtaining joint angles (ϕ_1, θ_2 through θ_6) for given end-effector position ($\underline{R}$) and orientation ($\underline{a}_6, \underline{S}_7$). The angle ϕ_1 is the rotation of the ground joint, which measures from the x-axis of the fixed coordinates (world coordinates) to the x-axis of the local coordinates of the ground joint, Fig. B-1. The parameters associated with the hypothetical joint can be calculated via a close-the-loop analysis.

The vector loop equation of the GE P60 robot is given by:

$$\underline{S}_{11}\underline{S}_1 - \underline{S}_{44}\underline{S}_4 + \underline{S}_{55}\underline{S}_5 + \underline{S}_{66}\underline{S}_6 + \underline{S}_{77}\underline{S}_7 + \underline{a}_{23}\underline{a}_{23} + \underline{a}_{34}\underline{a}_{34} + \underline{a}_{67}\underline{a}_{67} + \underline{a}_{71}\underline{a}_{71} = 0 \quad (\text{B.1})$$

A scalar product of equation (B.1) with the vector $\underline{S}_2$ yields

$$-\underline{S}_{44} + \underline{S}_{66}\underline{S}_6\underline{S}_2 + \underline{S}_{77}\underline{S}_7\underline{S}_2 + \underline{a}_{67}\underline{a}_{67}\underline{S}_2 + \underline{a}_{71}\underline{a}_{71}\underline{S}_2 = 0 \quad (\text{B.2})$$

Applying set 14 of the direction cosines of the spatial heptagon from Duffy (1981) to each term in equation (B.2) yields:

$$\underline{S}_6\underline{S}_2 = \underline{s}_{67}\underline{s}_7\underline{s}_1 - (\underline{s}_{71}\underline{c}_{67} + \underline{c}_{71}\underline{s}_{67}\underline{c}_7)\underline{c}_1$$

$$\underline{S}_7\underline{S}_2 = -\underline{s}_{71}\underline{c}_1$$

$$a_{71}s_2 = s_1$$

$$a_{67}s_2 = s_1c_7 + c_1s_7c_{71} \quad .$$

Substituting these results into equation (B.2) produces

$$-S_{44} + Ac_1 + Bs_1 = 0 \quad (B.3)$$

$$\text{where } A = a_{67}s_7c_{71} - S_{66}(s_{71}c_{67} + c_{71}s_{67}c_7) - S_{77}s_{71}$$

$$B = a_{67}c_7 + S_{66}s_{67}s_7 + a_{71} \quad .$$

Two values of angle θ_1 can be obtained from equation (B.3) using the half-tangent angle law.

$$\theta_1 = 2\tan^{-1}\left(\frac{B \pm (A^2 + B^2 - S_{44}^2)^{1/2}}{A + S_{44}}\right) \quad (B.4)$$

Set three of the subsidiary cosine formula yields the following relationship:

$$Z_{543} = Z_{71} \quad (B.5)$$

Expanding the term in each side produces two values of θ_5 :

$$\theta_5 = \pm \cos^{-1}(c_1(s_{71}c_{67} + c_{71}s_{67}c_7) - s_{67}s_7s_1) \quad (B.6)$$

The vectors $\underline{S}_2$, $\underline{S}_3$ and $\underline{S}_4$ are parallel, therefore, the subsidiary formulae for the spherical pentagon can be used. It follows that:

$$X_{17} = X_{56} \quad (B.7)$$

$$Y_{56} = -(\bar{X}_1s_7 + \bar{Y}_1c_7) \quad (B.8)$$

$$\bar{Z}_1 = Z_{56} \quad (B.9)$$

where

$$X_{56} = s_5c_6$$

$$X_{17} = s_1c_7 + c_{71}s_7$$

$$Y_{56} = c_{67}s_5s_6 + s_{67}c_5$$

$$\bar{X}_1 = s_1$$

$$\bar{Y}_1 = -c_{71}c_1$$

$$Z_{56} = s_{67}s_5s_6 - c_{67}c_5$$

$$Z_1 = -s_{71}c_1$$

Substitution of these terms into equations (B.7) through (B.9) and elimination of c_5 from equations (B.8) and (B.9) yields:

$$c_6 = \frac{s_1c_7 + c_{71}c_1s_7}{s_5} \quad (\text{B.10})$$

$$s_6 = \frac{c_{67}(c_{71}c_1c_7 - s_1s_7) - s_{67}s_{71}c_1}{s_5} \quad (\text{B.11})$$

A unique value of θ_6 can be obtained from equations (B.10) and (B.11). Substituting x- and y-components of set 14 of the direction cosines for the spatial heptagon into equation (B.1) yields:

$$\begin{aligned} -S_{44}X_{5671} + S_{55}X_{671} + S_{66}X_{71} + S_{77}X_1 + a_{23}c_2 + a_{34}W_{45671} + \\ a_{67}W_{71} + a_{71}c_1 = 0 \end{aligned} \quad (\text{B.12})$$

$$\begin{aligned} -S_{11}s_{12} - S_{44}Y_{5671} + S_{55}Y_{671} + S_{66}Y_{71} + S_{77}Y_1 - a_{23}s_2 - \\ a_{34}U_{456712}^* - a_{67}U_{712}^* + a_{71}U_{12}^* = 0 \end{aligned} \quad (\text{B.13})$$

where

$$X_{5671} = 0$$

$$X_{671} = s_6(c_7c_1 - c_{71}s_7s_1) + c_{67}c_6(s_7c_1 + c_{71}c_7s_1) - s_{67}c_6s_{71}s_1$$

$$X_{71} = s_{67}s_7c_1 + (s_{71}c_{67} + c_{71}s_{67}c_7)s_1$$

$$X_1 = s_{71}s_1$$

$$W_{45671} = c_{2+3}$$

$$W_{71} = c_1c_7 - s_1s_7c_{71}$$

$$Y_{5671} = 0$$

$$Y_{671} = -s_6s_{71}s_7 + c_{67}c_6s_{71}c_7 + s_{67}c_6c_{71}$$

$$Y_{71} = s_{71}s_{67}c_7 - c_{71}c_{67}$$

$$Y_1 = -c_{71}$$

$$U_{456712}^* = s_{2+3}$$

$$U_{712}^* = s_7 s_{71}$$

$$U_{12}^* = 0$$

One may see that the unknown terms are c_2 , s_2 , c_{2+3} and s_{2+3} . The equations (B.12) and (B.13) can be rearranged in the following form:

$$a_{23}c_2 + a_{34}c_{2+3} = -A \quad (B.14)$$

$$a_{23}s_2 + a_{34}s_{2+3} = B \quad (B.15)$$

where

$$A = S_{55}X_{671} + S_{66}X_{71} + S_{77}X_1 + a_{67}W_{71} + a_{71}c_1$$

$$B = -S_{11} + S_{55}Y_{671} + S_{66}Y_{71} + S_{77}Y_1 - a_{67}U_{712}^*$$

Summation of the squares of equations (B.14) and (B.15) yields:

$$a_{23}^2 + a_{34}^2 + 2a_{23}a_{34}c_3 = A^2 + B^2 \quad (B.16)$$

Two values of θ_3 are obtained:

$$\theta_3 = \pm \cos^{-1} \left(\frac{A^2 + B^2 - a_{23}^2 - a_{34}^2}{2 a_{23}a_{34}} \right) \quad (B.17)$$

Substitution of θ_3 into equations (B.14) and (B.15) leads to

$$(a_{23} + a_{34}c_3)c_2 - a_{34}s_3s_2 = -A \quad (B.18)$$

$$a_{34}s_3c_2 + (a_{23} + a_{34}c_3)s_2 = B \quad (B.19)$$

The unknowns c_2 and s_2 can be obtained from the above equations and the unique value of angle θ_2 results. Using the half-tangent angle law, the remaining unknown angle θ_4 can be obtained:

$$\tan \left(\frac{\theta_2 + \theta_3 + \theta_4}{2} \right) = \frac{X_{65} - X_1}{Y_{65} - Y_1} \quad (B.20)$$

After expanding the right side, a unique value of θ_4 can be obtained:

$$\theta_4 = 2 \tan^{-1} \left(\frac{s_{67}s_{65} + c_{67}s_5 - s_{71}s_1}{s_{67}c_6 + c_{71}} \right) - \theta_2 - \theta_3 \quad (B.21)$$

When the solutions are properly expanded, eight sets of solutions result. Two solution sets are meaningful due to the limitations on the range of joint actuators.

B.2 Forward Displacement Analysis of GE P60 Robot

Forward displacement analysis of a robot is used to obtain the position and orientation of the end-effector for a given set of joint angles. In order to obtain proper results set one of the direction cosine of the spatial heptagon should be used:

$$\underline{R}^1 = a_{23}a_{23}^1 + a_{34}a_{34}^1 - S_{44}S_4^1 + S_{55}S_5^1 + S_{66}S_6^1 + a_{67}a_{67}^1 \quad (B.22)$$

Expansion of the components are as follows:

$$R_x^1 = a_{23}c_2 + a_{34}c_{2+3} + S_{55}s_{2+3+4} + S_{66}s_5c_{2+3+4} + a_{67}(s_6s_{2+3+4} + c_5c_6c_{2+3+4})$$

$$R_y^1 = S_{44} + S_{66}c_5 - a_{67}s_5c_6$$

$$R_z^1 = a_{23}s_2 + a_{34}s_{2+3} - S_{55}c_{2+3+4} + S_{66}s_5s_{2+3+4} + a_{67}(c_5c_6s_{2+3+4} - s_6c_{2+3+4})$$

The vector $\underline{R}^1$ is in local coordinates defined by set one, therefore, it should be transformed to world coordinates:

$$\underline{R}^{world} = M\underline{R}^1 \quad (B.23)$$

where M is the rotation matrix given by:

$$M = \begin{bmatrix} \cos\phi_1 & -\sin\phi_1 & 0 \\ \sin\phi_1 & \cos\phi_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (B.24)$$

The orientation vectors $\underline{a}_{67}^1(a_{67x}^1, a_{67y}^1, a_{67z}^1)$ and $\underline{S}_7^1(S_{7x}^1, S_{7y}^1, S_{7z}^1)$ are expressed as follows:

$$a_{67x}^1 = s_6s_{2+3+4} + c_5c_6s_{2+3+4}$$

$$a_{67y}^1 = -s_5c_6$$

$$a_{67z}^1 = -s_6c_{2+3+4} + c_5c_6s_{2+3+4}$$

$$S_{7x}^1 = s_{67}s_6c_5c_{2+3+4} + c_{67}s_5c_{2+3+4} - s_{67}c_6c_{2+3+4}$$

$$S_{7y}^1 = -s_{67}s_6s_5 + c_{67}c_5$$

$$S_{7z}^1 = s_{67}s_6c_5s_{2+3+4} + c_{67}s_5s_{2+3+4} + s_{67}c_6s_{2+3+4} \quad .$$

The orientation vectors in world coordinates can be obtained as follows:

$$\underline{a}_{67}^{\text{world}} = M\underline{a}_{67}^1 \quad (\text{B.25})$$

$$\underline{s}_7^{\text{world}} = M\underline{s}_7^1 \quad . \quad (\text{B.26})$$

The forward displacement analysis is useful when examining if the data to the reverse displacement analysis is valid. The input position and orientation of the end-effector is compared with the results of the forward displacement analysis. If they do not match, the out-of-the-reachable-workspace situation occurs.

REFERENCES

- Angeles, J., Alivizotos, A., and Zsombor-Murray, P.J., 1986, "The Synthesis of Smooth Trajectories for Pick-and-Place Operation," IEEE Proc. Int. Conf. on Systems, Man, and Cybernetics, Atlanta, pp.666-670.
- Avnaim, F., Boissonnat, J.D., and Faverjon, B., 1988, "A Practical Exact Motion Planning Algorithm for Polygonal Objects Amidst Polygonal Obstacles," IEEE Proc. Int. Conf. on Robotics and Automation, Philadelphia, pp.1656-1661.
- Bajaj, C., and Kim, M.S., 1987, "Generation of Configuration Space Obstacles: The Case of Moving Algebraic Curves," IEEE Proc. Int. Conf. on Robotics and Automation, Raleigh, N.C., pp.979-984.
- Bajaj, C., and Kim, M.S., 1988, "Generation of Configuration Space Obstacles: The Case of a Moving Sphere," IEEE J. of Robotics and Automation, Vol. 4, No.1, pp.94-99.
- Basta, R.A., Mehrota, R., and Varanasi, M.R., 1988, "Collision Detection for Planning Collision-Free Motion of Two Robot Arms," IEEE Proc. Int. Conf. on Robotics and Automation, pp.638-640.
- Bézier, P., 1972, Numerical Control-Mathematics and Application, Translated by A.R. Forrest and A.F. Pankhurst, John Wiley & Sons, New York.
- Bézier, P., 1974, "Mathematical and Practical Possibilities of UNISURF," in Computer Aided Geometric Design, Edited by R.E. Barnhill and R.F. Riesenfeld, Academic Press, New York.
- Borenstein, J., and Koren, Y., 1988, "Obstacle Avoidance with Ultrasonic Sensors," IEEE J. of Robotics and Automation, Vol. 4, No.2, pp.213-218.
- Boyse, J.W., 1979, "Interference Detection Among Solids and Surfaces," Comm. ACM, Vol.22, No.1, pp.3-9.
- Brooks, R.A., 1983a, "Solving the Find-Path Problem by Good Representation of Free Space," IEEE Trans. on Systems, Man and Cybernetics, Vol.SMC-13, No.3, pp.190-197.
- Brooks, R.A., 1983b, "Planning Collision-Free Motions for Pick-and-Place Operations," Int. J. of Robotics Research, Vol.2, No.4, pp.19-44.

- Brooks, R.A., 1986a, "A Robust Layered Control System for a Mobile Robot," IEEE J. Robotics and Automation, Vol. RA-2, No.1, pp.14-23.
- Brooks, R.A., 1986b, "A Layered Intelligent Control System for a Mobile Robot," Robotics Research, The Third International Symposium, Edited by O.D. Faugeras and George Giralt, MIT Press, Cambridge, Held in Chateau Monvillargene (Gouvieux), France, pp.365-372.
- Brooks, R.A., 1987, "A Hardware Retargetable Distributed Layered Architecture for Mobile Robot Control," IEEE Proc. Int. Conf. on Robotics and Automation, Raleigh, N.C., pp.106-110.
- Brooks, R.A., and Lozano-Pérez, T., 1985, "A Subdivision Algorithm in Configuration Space for Findpath with Rotation," IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-15, No.2, pp.224-233.
- Burks, B.L., Barnett, D.L., Killough, S.M., Mann, R.C., and Pin, F.G., 1987, "Dealing with Unanticipated Events in Autonomous Navigation," IEEE Conf. on Decision and Control, pp.1808-1813.
- Gao, Z.L., Huang, Y., and Hall, E.L., 1988, "Region Filling Operations with Random Obstacle Avoidance for Mobile Robots," J. of Robotic Systems, Vol.5, No.2, pp.87-102.
- Chatila, R., 1986, "Mobile Robot Navigation: Space Modeling and Decisional Processes," Robotics Research, The Third International Symposium, Edited by O.D. Faugeras and George Giralt, MIT Press, Cambridge, Held in Chateau Monvillargene (Gouvieux), France, pp.373-378.
- Chattergy, R., 1985, "Some Heuristics for the Navigation of a Robot," Int. J. Robotics Research, Vol.4, No.1, pp.59-66.
- Chen, Y.C., and Vidyasagar, M., 1987, "Some Qualitative Results on the Collision-Free Joint Space of a Planar n-DOF Linkage," IEEE Proc. Int. Conf. on Robotics and Automation, Raleigh, N.C., pp.1623-1630.
- Chen, Y.C., and Vidyasagar, M., 1988, "Optimal Trajectory Planning for Planar n-Link Revolute Manipulators in the Presence of Obstacles," IEEE Proc. Int. Conf. on Robotics and Automation, Philadelphia, pp.202-208.
- Cheung, E., and Lumelsky, V., 1988, "Motion Planning for Robot Arm Manipulators with Proximity Sensing," IEEE Proc. Int. Conf. on Robotics and Automation, Philadelphia, pp.740-745.
- Chien, Y.P., Koivo, A.J., and Lee, B.H., 1988, "On-line Generation of Collision Free Trajectories for Multiple Robots," IEEE Proc. Int. Conf. on Robotics and Automation, Philadelphia, pp.209-214.
- Crane, C.D., III, 1987, "Motion Planning and Control of Robot Manipulators via Application of a Computer Graphics Animated Display," Ph.D. Dissertation, University of Florida.

Crowley, J.L., 1985, "Navigation for an Intelligent Mobile Robot," IEEE J. Robotics and Automation, Vol. RA-1, No.1, pp.31-41.

Daily, M., Harris, J., Keirsey, D., Olin, K., Payton, D., Reiser, K., Rosenblatt, J., Tseng, D., and Wong, V., 1988, "Autonomous Cross-Country Navigation with the ALV," IEEE Proc. Int. Conf. on Robotics and Automation, Philadelphia, pp.718-726.

Davidson, J.K., and Hunt, K.H., 1987, "Robot Workspace of a Tool Plane: Part 1--A Ruled Surface and Other Geometry," ASME Journal of Mechanisms, Transmissions, and Automation in Design, Vol. 109, pp.50-60.

Davidson, J.K., and Pingali, P., 1987, "Robot Workspace of a Tool Plane: Part 2--Computer Generation and Selected Design Condition for Dexterity," ASME Journal of Mechanisms, Transmissions, and Automation in Design, Vol. 109, pp.61-71.

Duffy, J., 1981, Analysis of Mechanisms and Robot Manipulator, John Wiley & Sons, New York.

Dupont, P.E., and Derby, S., 1988, "A Two Phase Path Planning Planning Algorithm for Robots with Six or more Joints," Trends and Developments in Mechanisms, Machines, and Robotics. ASME 20th Biennial Design Automation Conference, Orlando, Vol. 3, pp.415-428.

Dupont, P.E., and Derby, S., 1988, "A Simple Heuristic Path Planner for Redundant Robots," Trends and Developments in Mechanisms, Machines, and Robotics. ASME 20th Biennial Design Automation Conference, Orlando, Vol.3, pp.415-428.

Elfes, A., and Matthies, L., 1987, "Sensor Integration for Robot Navigation: Combining Sonar and Stereo Range Data in a Grid-Based Representation," IEEE Conf. on Decision and Control, pp.1802-1807.

Espiau, B., and Boulic, R., 1986, "Collision Avoidance for Redundant Robots with Proximity Sensors," Robotics Research, The Third International Symposium, Edited by O.D. Faugeras and George Giralt, MIT Press, Cambridge, Held in Chateau Monvillargene (Gouvieux), France, pp.243-251.

Faux, I.D., and Pratt, M.J., 1979, Computational Geometry for Design and Manufacture, John Wiley & Sons, New York.

Foley, J.D., and Van Dam, A., 1982, Fundamentals of Interactive Computer Graphics, Addison Wesley, Reading, Massachusetts.

Freund, E., and Hoyer, H., 1985, "Collision Avoidance in Multi-Robot Systems," Robotics Research, The Second International Symposium, Edited by Hideo Hanafusa and Hirochika Inoue, MIT Press, Cambridge, Mass., Held in Kyoto Japan, pp.135-146.

Freund, E., and Hoyer, H., 1986, "On the On-line Solution of the Findpath Problem in Multi-Robot Systems," Robotics Research, The Third

International Symposium, Edited by O.D. Faugeras and George Giralt, MIT Press, Cambridge, Held in Chateau Monvillargene (Gouvieux), France, pp.253-262.

Freund, E., and Hoyer, H., 1988, "Real-Time Pathfinding in Multirobot Systems Including Obstacle Avoidance," Int. J. of Robotics Research, Vol.7, No.1, pp.42-79.

Fugimura, K., and Samet, H., 1989, "A Hierarchical Strategy for Path Planning Among Moving Obstacles," IEEE Proc. Int. Conf. on Robotics and Automation, Scottsdale (Arizona), pp.61-69.

Fugimura, K., and Samet, H., 1988, "Path Planning Among Moving Obstacles Using Spatial Indexing," IEEE Proc. Int. Conf. on Robotics and Automation, Philadelphia, pp.1662-1667.

Geering, H.P., Guzzella, L., Hepner, S.A.R., and Onder, C.H., "Time-Optimal Motions of Robotics in Assembly Task," IEEE Trans. Automatic Control, Vol. AC-31, No.6, pp.512-518

Gex, W. T., and Campbell, N. L. 1987, "Local Free Space Mapping and Path Guidance," IEEE Proc. Int. Conf. on Robotics and Automation, Raleigh, N.C., pp.424-431.

Giralt, G., Chatila, R., and Vaisset, M., 1984, "An Integrated Navigation and Motion Control System for Autonomous Multisensory Mobile Robots," Robotics Research, The First International Symposium, Edited by Michael Brady and Richard Paul, MIT Press, Cambridge, Mass., Held at Bretton Woods, N.H., pp.191-214.

Giralt, G., Sobek, R., and Chatila, R., 1979, "A Multi-Level Planning and Navigation System for a Mobile Robot; A First Approach to HILARE," Proc. 6th Int. Conf. on Artificial Intelligence, Tokyo, Japan, pp.335-337.

Gilbert, E.G., and Johnson, D.W., 1985, "Distance Functions and Their Application to Robot Path Planning in the Presence of Obstacles," IEEE J. of Robotics and Automation, Vol. RA-1, No.1, pp.21-30.

Goult, R.J., Hoskins, R.F., Milner, J.A., and Pratt, M.J., 1973, Applicable Mathematics. A Course for Scientists and Engineers, Macmillan Co, London.

Gouzènes, L. 1984, "Strategies for Solving Collision-Free Trajectories Problems for Mobile and Manipulator Robots," Int. J. of Robotics Research, Vol.3, No.4, pp.51-65.

Gupta, K.C., and Roth, B., 1982, "Design Considerations for Manipulator Workspace," ASME J. Mechanical Design, Vol. 104, pp.704-711.

Gupta, K.C., 1987, "On the Nature of Workspace," in The Kinematics of Robot Manipulator, edited by J.M. McCarthy, MIT Press, Cambridge.

- Hopcroft, J.E., Schwartz, J.T., and Sharir, M., 1984, "On the Complexity of Motion Planning for Multiple Independent Object; PSPACE-hardness of the 'Warehouseman's Problem'," *Int. J. Robotics Research*, Vol.3, No.4, pp.76-88.
- Hsu, P., Hauser, J., and Sastry, S., 1988, "Dynamic Control of Redundant Manipulators," *IEEE Proc. Int. Conf. on Robotics and Automation*, Philadelphia, pp.183-187.
- Ish-Shalom, J., 1987, "The Funnel Algorithm and Task Level Robot Control," *IEEE Proc. Int. Conf. on Robotics and Automation*, Raleigh, N.C., pp.25-32.
- Isik, C., 1985, "Knowledge-Based Motion Control of an Intelligent Mobile Autonomous System," Ph.D. Dissertation, University of Florida.
- Iyengar, S.S., Jorgensen, C.C., Rao, S.V., and Weisbin, C.R., 1985, "Learned Navigation Paths for a Robot in Unexplored Terrain," *IEEE, Int. Conf. on Artificial Intelligence Application*, Miami Beach, FL, pp.148-155.
- Javis, R.A., 1985, "Projection Derived Space Cube Scene Models for Robotic Vision and Collision-Free Planning," *Robotics Research, The Second International Symposium*, Edited by Hideo Hanafusa and Hirochika Inoue, MIT Press, Cambridge, Mass., Held in Kyoto Japan, pp.403-410.
- Jun, S., and Shin, K.G., 1988, "A Probabilistic Approach to Collision-Free Path Planning," *IEEE Proc. Int. Conf. on Robotics and Automation*, Philadelphia, pp.220-225.
- Kambhampati, S., and Davis, L.S., 1986, "Multiresolution Path Planning for Mobile Robot," *IEEE J. of Robotics and Automation*, Vol. RA-2, No.3, pp.135-145.
- Kanayama, Y., 1988, "Least Cost Paths with Algebraic Cost Functions," *IEEE Proc. Int. Conf. on Robotics and Automation*, Philadelphia, pp.75-80.
- Kanayama, Y., and Miyake, N., 1986, "Trajectory Generation for Mobile Robots," *Robotics Research, The Third International Symposium*, Edited by O.D. Faugeras and George Giralt, MIT Press, Cambridge, Held in Chateau Monvillargene (Gouvieux), France, pp.333-340.
- Kanayama, Y., and Yuta, S., 1988, "Vehicle Path Specification by a Sequence of Straight Lines," *IEEE J. of Robotics and Automation*, Vol.4, No.3, pp.265-276.
- Kant, K., and Zucker, S., 1988, "Planning Collision-Free Trajectories in Time-Varying Environments: A Two-Level Hierarchy," *IEEE Proc. Int. Conf. on Robotics and Automation*, Philadelphia, pp.1644-1649.
- Keirsey, D.M., Mitchell, J.S.B., Payton, D.W., and Preyss, E.P., 1984, "Multilevel Path Planning for Autonomous Vehicles," *Society of Photo-*

Optical Instrumentation Engineers Vol.485, Applications of Artificial Intelligence, Arlington, Virginia, pp.133-137.

Khatib, O., 1986, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," Int. J. of Robotic Research, Vol.5, No.1, pp.90-98.

Khatib, O., 1986, "The Operational Space Formulation in the Analysis, Design, and Control of Robot Manipulators," Robotics Research, The Third International Symposium, Edited by O.D. Faugeras and George Giralt, MIT Press, Cambridge, Held in Chateau Monvillargene (Gouvieux), France, pp.263-270.

Khosla, P., and Volpe, R., 1988, "Superquadric Artificial Potential for Obstacle Avoidance and Approach," IEEE Proc. Int. Conf. on Robotics and Automation, Philadelphia, pp.1778-1784.

Kim, B.K., and Shin, K.G., 1985, "Minimum-Time Path Planning for Robot Arms and Their Dynamics," IEEE Trans. Systems, Man and Cybernetics, Vol. SMC-15, No.2, pp.213-223.

Koditschek, D.E., 1987, "Exact Robot Navigation by Means of Potential Functions: Some Topological Considerations," IEEE Proc. Int. Conf. on Robotics and Automation, Raleigh, N.C., pp.1-6.

Kumar, A., and Waldron, K.J., 1981, "The Workspace of a Mechanical Manipulator," ASME J. Mechanical Design, Vol. 103, pp.665-672.

Lee, B.H., 1986, "Wrist Collision Avoidance of Two Robots: A Collision Map and Time Scheduling Approach," IEEE Proc. of 25th conf. on Decision and Control, Athens, Greece, pp.429-433.

Lee, B.H., and Chien, Y.P., 1987, "Time-Varying Obstacle Avoidance for Robot Manipulators: Approaches and Difficulties," IEEE Proc. Int. Conf. on Robotics and Automation, Raleigh, N.C., pp.1610-1615.

Lee, D.T., and Preparata, F.P., 1984, "Computational Geometry--A Survey," IEEE Trans. on Computers. Vol. c-33, No. 12, pp.1072-1101.

Leon, S.J., 1986, Linear Algebra with Applications, Macmillan Co., New York.

Lipkin, H., Duffy, J., 1985, "A Vector Analysis of Robot Manipulator," in Recent Advances in Robotics, Edited by G. Beni and S. Hackwood, John Wiley & Sons, New York.

Lovell, G.M., III, 1983, "Quantification of Dexterity for Some Planar and Spatial Robot," Master's Thesis, University of Florida.

Lozano-Pérez, T., 1981, "Automatic Planning of Manipulator Transfer Movements," IEEE Trans. Systems, Man, and Cybernetics, Vol. SMC-11, No.10, pp.681-698.

Lozano-Pérez, T., 1982, "Task Planning," in Robot Motion: Planning and Control, Edited by M. Brady, J.M. Hollerbach, T.L. Johnson, T. Lozano-Pérez and M.T. Mason, MIT Press, Cambridge, pp.473-498.

Lozano-Pérez, T., 1983, "Spatial Planning: A Configuration Space Approach," IEEE Trans. Computers, Vol. C-32, No.2, pp.108-120.

Lozano-Pérez, T., 1986, "Motion Planning for Simple Robot Manipulators," Robotics Research, The Third International Symposium, Edited by O.D. Faugeras and George Giralt, MIT Press, Cambridge, Held in Chateau Monvillargene (Gouvieux), France, pp.133-140.

Lozano-Pérez, T., 1987, "A Simple Motion-Planning Algorithm for General Robot Manipulators," IEEE J. of Robotics and Automation, Vol. RA-3, No.3, pp.224-238.

Lozano-Pérez, T., Jones, J.L., Mazer, E., O'Donnell, P.A., and Grimson, W.E.L., 1987, "Handy: A Robot System that Recognizes, Plans and Manipulates," IEEE Proc. Int. Conf. on Robotics and Automation, Raleigh, N.C., pp. 843-849.

Lozano-Pérez, T., and Wesley, M.A., 1979, "An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles," Comm. ACM, Vol.22, No.10, pp.560-570.

Lumelsky, V.J., 1985, "Effect of Robot Kinematics on Motion Planning in Unknown Environment," IEEE Proc. 24th Conf. Decision and Control, Fort Lauderdale, FL, pp.338-343.

Lumelsky, V.J., 1987a, "Gross Motion Planning for a Simple 3D Articulated Robot Arm Moving Amidst Unknown Arbitrarily Shaped Obstacles," IEEE Proc. Int. Conf. on Robotics. and Automation, Raleigh, N.C., pp.1929-1934.

Lumelsky, V.J., 1987b, "Dynamic Path Planning for a Planar Articulated Robot Arm Moving Amidst Unknown Obstacles," Automatica, Vol.25, No.5, pp.551-570.

Lumelsky, V.J., 1987c, "Effect of Kinematics on Motion Planning for Planar Robot Arms Moving Amidst Unknown Obstacles," IEEE J. of Robotics and Automation, Vol. RA-3, No.3, pp.207-223.

Lumelsky, V.J., 1987d, "Algorithmic Issues of Sensor-Based Robot Motion Planning," IEEE Proc. 26th Conf. Decision and Control, Los Angeles, pp.1796-1801.

Lumelsky, V.J., and Stepanov, A.A., 1986, "Dynamic Path Planning for a Mobile Automation with Limited Information on the Environment," IEEE, Trans. Automatic Control, Vol. AC-31, No.11, pp.1058-1063.

Lumelsky, V.J., and Skewis, T., 1988, "A Paradigm for Incorporating Vision in the Robot Navigation Function," IEEE Proc. Int. Conf. on Robotics and Automation, Philadelphia, pp.734-739.

Maciejewski, A.A., and Klein, C.A., 1985, "Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environments," Int. J. Robotics Research, Vol.4, No.3, pp.109-117.

Matties, L., and Elfes, A., 1988, "Integration of Sonar and Stereo Range Data Using a Grid-Based Representation," IEEE Proc. Int. Conf. on Robotics and Automation, Philadelphia, pp.727-733.

Meyer, W., and Benedict, P., 1988, "Path Planning and the Geometry of Joint Space Obstacles," IEEE Proc. Int. Conf. on Robotics and Automation, Philadelphia, pp.215-219.

Mitchell, J.S.B., 1984, "An Autonomous Vehicle Navigation Algorithm," Society of Photo-Optical Instrumentation Engineers, Vol.485, Applications of Artificial Intelligence, pp.153-158.

Mitchell, J.S.B., and Keirsey, D.M., 1984, "Planning Strategic Paths through variable terrain data," Society of Photo-Optical Instrumentation Engineers, Vol.485, Applications of Artificial Intelligence, pp.172-179.

Missiaen, L., Lecluyse, H., and Bruynooghe, M., 1987, "Navigation Algorithm for a Mobile Robot Using Ultrasonic Sensors," Artificial Intelligence and Information--Control Systems of Robots, Edited by I. Plander, Elsevier Science Pub, New York, pp.63-75.

Montgomery, M., Gaw, D., and Meystel, A., 1987, "Navigation Algorithm for a Nested Hierarchical System of Robot Path Planning Among Polyhedral Obstacles," IEEE Proc. Int. Conf. on Robotics and Automation, Philadelphia, pp.1616-1622.

Moravec, H. P., 1980, Robot Rover Visual Navigation, UMI Research Press, Ann Arbor.

Nagata, T., Honda, K., and Teramoto, Y., 1988, "Multirobot Plan Generation in a Continuous Domain: Planning by Use of Plan Graph and Avoiding Collisions Among Robots," IEEE J. Robotics and Automation, Vol.4, No.1, pp.2-13.

Nasr, H., and Bhanu, B., 1988, "Landmark Recognition for Autonomous Mobile Robot," IEEE Proc. Int. Conf. on Robotics and Automation, Philadelphia, pp.1218-1223.

Newman, W.S., and Hogan, N. 1987, "High Speed Robot Control and Obstacle Avoidance Using Dynamic Potential Function," IEEE Proc. Int. Conf. on Robotics and Automation, Raleigh, N.C., pp.14-24.

Nilsson, N., 1980, Principles of Artificial Intelligence, Springer-Verlag, Berlin, pp.76-84.

Paul, R.P., Shimano, B., and Mayer, G.E., 1981, "Kinematic Control Equations for Simple Manipulators," IEEE Trans. on Systems, Man and Cybernetics Vol. SMC-11, No. 6., pp.449-455.

Preparata, F.P., and Shamos, M.I., 1985 Computational Geometry, Springer-Verlag, Berlin.

Rao, N.S.V., Iyengar, S.S., and deSaussur, G., 1988, "The Visit Problem: Visibility Graph-Based Solution," IEEE Proc. Int. Conf. on Robotics and Automation, Philadelphia, pp.1650-1655.

Red, W.E., Troung-Cao, H.V., and Kim, K.H., 1987, "Robot Path Planning in Three-Dimensions Using the Direct Subspace," ASME J. Dynamic Systems, Measurement, and Control, Vol.109, pp.238-244.

Richbourg, R.F., Rowe, N.C., Zyda, M.J., and McGee, R.B., 1987, "Solving Global Two-Dimensional Routing Problems Using Snell's Law and A* Search," IEEE Proc. Int. Conf. on Robotics and Automation, Raleigh, N.C., pp.1631-1636.

Rimon, E., and Koditschek, D.E., 1988, "Exact Robot Navigation using Cost Functions: The Case of Distinct Spherical Boundaries in E," IEEE Proc. Int. Conf. on Robotics and Automation, Philadelphia, pp.1791-1796.

Rogers, D.F., 1985, Procedural Elements for Computer Graphics, McGraw-Hill, New York.

Rogers, D.F., and Adams, J.A., 1976, Mathematical Elements for Computer Graphics, McGraw-Hill, New York.

Schwartz, J.T., and Sharir, M., 1983a, "On the Piano Movers' Problem: I. The Case of a Two-Dimensional Rigid Polygonal Body Moving Amidst Polygonal Barriers," Comm. Pure and Applied Mathematics, Vol. 36, pp.345-398.

Schwartz, J.T., and Sharir, M., 1983b, "On the Piano Movers' Problem: II. General Techniques for Computing Topological Properties of Real Algebraic Manifolds," Adv. Applied Mathematics, Vol. 4, pp.298-351.

Schwartz, J.T., and Sharir, M., 1983c, "On the Piano Movers' Problem: III. Coordinating the Motion of Several Independent Bodies: The Special Case of Circular Bodies Moving Amidst Polygonal Barriers," Int. J. Robotics Research, Vol.2, No.3, pp.46-75.

Schwartz, J.T., and Sharir, M., 1984, "On the Piano Movers' Problem: V. The Case of a Rod Moving in Three-dimensional Space Amidst Polyhedral Obstacles," Comm. Pure and Applied Mathematics, Vol. 37, pp.815-848.

Sciavicco, L., and Siciliano, B., 1988, "A Solution Algorithm to the Inverse Kinematic Problem for Redundant Manipulators," IEEE J. Robotics and Automation, Vol.4, No. 4, pp.403-410.

Seida, S., and Morgenthaler, D.G., 1987, "Vision-Based Road Following in the Autonomous Land Vehicle," IEEE Proc. 26th Conf on Decision and Control, Los Angeles, pp.1814-1819.

Sharir, M., and Ariel-Sheffi, E., 1984, "On the Piano Movers' Problem: IV. Various Decomposable Two-Dimensional Motion-Planning Problems," *Comm. Pure and Applied Mathematics*, Vol. 37, pp.479-493.

Singh, J.S., and Wagh, M.D., 1987, "Robot Path Planning Using Intersecting Convex Shapes: Analysis and Simulation," *IEEE J. Robotics and Automation*, Vol. RA-3, No.2, pp.101-108.

Steele, J.P.H., and Starr, G.P., 1987, "Path Planning and Heuristics for Mobile Robots in Real World Environments," *IEEE Proc. Int. Conf. on Robotics and Automation*, Raleigh, N.C., pp.365-369.

Sugimoto, K., and Duffy, J. 1981, "Determination of Extreme Distances of a Robot Hand--Part I. A General Theory," *ASME Journal of Mechanical Design*, Vol. 103, No.3, pp.631-636.

Sugimoto, K., Duffy, J., and Hunt, K.M. 1982, "Special Configurations of Spatial Mechanisms and Robot Arms," *Mechanism and Machine Theory*, Vol. 17, No. 2, pp.119-132.

Tompson, A.M., 1977, "The Navigation System of the JPL Robot," *Proc. 5th Int. Joint Conf. on Artificial Intelligence*, Cambridge, Mass., pp.749-757.

Tournassoud, P., and Jehl, O., 1988, "Motion Planning for a Mobile Robot with a Kinematic Constraint," *IEEE Proc. Int. Conf. on Robotics and Automation*, pp.1785-1790.

Udupa, S.M., 1977, "Collision Detection and Avoidance in Computer Controlled Manipulators," *Proc. 5th Int. Joint Conf. on Artificial Intelligence*, Cambridge, Mass., pp.737-748.

Volpe, R., and Khosla, P., 1987, "Artificial Potentials with Elliptical Isopotential Contours for Obstacle Avoidance," *IEEE Proc. 26th Conf. Decision and Control*, Los Angeles, pp.180-185.

Waldron K.J., and Stevenson, E.N., Jr., 1979, "Elimination of Branch, Grashof and Order Defects in Path-Angle Generation and Function Generation Synthesis," *ASME Journal of Mechanical Design*, Vol. 101, No.3, pp.428-437.

Wang, S., and Waldron, K.J., 1986, "A Study of the Singular Configurations of Serial Manipulator," *ASME Journal of Mechanisms, Transmissions and Automation in Design*, Vol. 109, pp.14-20.

Waxman, A.M., LeMoigne, J.J., Davis, L.S., Srinivasan, B., Kushner, T.R., Liang, E., and Siddalingaiah, T., 1987, "A Visual Navigation System for Autonomous Land Vehicles," *IEEE J. Robotics and Automation*, Vol. RA-3, No.2, pp.124-141.

Wilfong, G.T., 1988, "Motion Planning for an Autonomous Vehicle," *IEEE Proc. Int. Conf. on Robotics and Automation*, Philadelphia, pp.529-533.

Wu, Y.F., Widmayer, P., Schlag, M.D.F., and Wong, C.K., "Rectilinear Shortest Path and Minimum Spanning Trees in the Presence of Rectilinear Obstacles," IEEE Trans. Computer, Vol. C-36, No.3, pp.321-331.

Yeung, D.Y., and Bekey, G.A., 1987, "A Decentralized Approach to the Motion Planning Problem for Multiple Mobile Robots," IEEE Proc. Int. Conf. on Robotics and Automation, Raleigh, N.C., pp.1779-1784.

Young, L., 1986, "A Theory of Articulation," Ph.D. Dissertation, University of Florida.

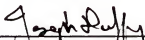
Young, L., and Duffy, J., 1987, "A Theory for the Articulation of Planar Robots: Part 1--Kinematic Analysis for the Flexure and the Parallel Operation of Robots," ASME J. Mechanisms, Transmissions, and Automation in Design, Vol. 109, pp.29-36.

BIOGRAPHICAL SKETCH

Ihn Namgung was born August 27, 1959, at Heongsung, Republic of Korea. He received the Bachelor of Engineering degree in 1981, and the Master of Science in Mechanical Engineering degree in 1983 from Ajou University. After serving his military duty as an army officer in 1984, he continued his graduate studies at the University of Florida in 1985.

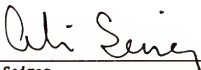
In December, 23, 1988, he married the former Yearnok Yoon.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



Joseph Duffy, Chairman
Graduate Research Professor of
Mechanical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



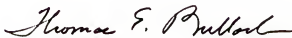
Ali Seireg
Professor of Mechanical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



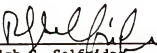
Carl D. Crane III
Assistant Professor of
Mechanical Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



Thomas E. Bullock
Professor of Electrical Engineering

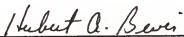
I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



Ralph G. Selfridge
Professor of Computer and
Information Science

This dissertation was submitted to the Graduate Faculty of the College of Engineering and to the Graduate School and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

December 1989



for Winfred M. Phillips
Dean, College of Engineering

Madelyn M. Lockhart
Dean, Graduate School